Promting

Step 3. RICCE prompting

Role:

You are an expert web design and accessibility designer.

Objective:

Create a landing page to present an accessibility course.

Context:

The code generated must have accessibility features such as complying with ALL WCAG 2.2 guidelines, ‘Making Content Usable for People with Cognitive and Learning Disabilities’ guidelines, and incorporating WAI-ARIA code into the appropriate elements. In addition, considerations for easy language, keyboard navigation, and landmarks that allow for zone navigation.

List of WCAG 2.2 guidelines (the WCAG 2.2 table is included directly)

Criterio WCAG Nivel Versión

1.3.1: Información y relaciones A 2.0

1.3.2: Secuencia significativa A 2.0 / 2.1

3.1.1: Idioma de la página A 2.0

3.1.2: Idioma de partes AA 2.0

4.1.1: Parsing (Análisis) A 2.0

4.1.2: Nombre, función, valor A 2.0

4.1.3: Mensajes de estado AA 2.1

2.4.1: Omitir bloques A 2.0 / 2.1

2.4.5: Múltiples maneras AA 2.0 / 2.1

3.2.3: Navegación consistente AA 2.0

3.2.4: Identificación consistente AA 2.0

3.2.6: Ayuda consistente A 2.2

2.4.4: Propósito del enlace (en contexto) A 2.0 / 2.1

2.4.3: Orden del foco A 2.0 / 2.1

2.4.7: Foco visible AA 2.0

2.4.11: Foco no oculto (mínimo) AA 2.2

1.4.13: Contenido en foco o al pasar el mouse AA 2.1

2.1.1: Teclado A 2.0 / 2.1

2.1.2: Sin trampa de teclado A 2.0 / 2.1

2.1.4: Atajos de teclado A 2.1

2.2.1: Tiempo ajustable A 2.0 / 2.2

2.2.2: Pausar, detener, ocultar A 2.0

2.4.2: Título de la página A 2.0 / 2.1

2.4.6: Encabezados y etiquetas AA 2.0 / 2.1

1.3.3: Características sensoriales A 2.0

1.4.1: Uso del color A 2.0

1.4.3: Contraste (mínimo) AA 2.0

1.4.11: Contraste de elementos no textuales AA 2.1

1.3.4: Orientación AA 2.1

1.4.10: Reflujo (Reflow) AA 2.1

1.4.12: Espaciado de texto AA 2.1

2.5.8: Tamaño del objetivo (mínimo) AA 2.2

1.4.4: Redimensionar texto AA 2.0

2.5.1: Gestos de puntero A 2.1

2.5.2: Cancelación de puntero A 2.1

2.5.4: Activación por movimiento A 2.1

2.5.7: Movimientos de arrastre AA 2.2

3.2.1: Al recibir el foco A 2.0

3.2.2: Al ingresar datos A 2.0

1.3.5: Identificar propósito de entrada AA 2.1

2.5.3: Etiqueta en nombre A 2.1

3.3.1: Identificación de errores A 2.0

3.3.2: Etiquetas o instrucciones A 2.0

3.3.3: Sugerencia de errores AA 2.0

3.3.4: Prevención de errores (legal, financiero, datos) AA 2.0

3.3.7: Entrada redundante A 2.2

3.3.8: Autenticación accesible (mínimo) AA 2.2

1.1.1: Contenido no textual A 2.0

1.4.5: Imágenes de texto AA 2.0

1.2.1: Solo audio o solo vídeo (pregrabado) A 2.0

1.2.2: Subtítulos (pregrabados) A 2.0

1.2.3: Descripción de audio o alternativa A 2.0

1.2.4: Subtítulos (en vivo) AA 2.0

1.2.5: Descripción de audio (pregrabado) AA 2.0

1.4.2: Control de audio A 2.0

2.3.1: Tres destellos o menos A 2.0

Accessibility website sources:

WCAG 2.2 guidelines: https://www.w3.org/TR/WCAG22/

COGA guidelines: https://www.w3.org/TR/coga-usable/

https://www.w3.org/TR/wai-aria/

https://www.w3.org/WAI/ARIA/apg/patterns/

It should also include usability aspects.

Lista de heuristicas de toni

1 1- Visibilidad y estado del sistema / Visibility and system state

1.1 La aplicación incluye de forma visible el título de la página, de la sección o del sitio?

Does the application include a visible title page, section or site?

1.2 El usuario sabe en todo momento dónde está?

Does the user always know where he is located?

1.3 El usuario sabe en todo momento qué está haciendo el sistema o aplicación?

Does the user always know what the system or application is doing?

1.4 Los enlaces están claramente definidos?

Are the links clearly defined?

1.5 Todas las acciones pueden verse directamente? (Sin requerir acciones adicionales)

Ejemplo: Si nos encontramos dentro de un proceso de compra, todas las acciones necesarias en cada uno de los pasos de este proceso se debe poder ver y entender directamente, sin necesidad de realizar acciones adicionales.

Can all actions be visualized directly? (No other actions are required)

Example: If we are within a purchase process, all the necessary actions in each of the steps of this process must be seen and understood directly, without the need for additional actions.

2 "2 - Connexión entre el sistema y el mundo real, uso de metáforas y objetos humanos /

Connection between the system and the real world, metaphor usage and human objects"

2.1 La información aparece de una manera lógica para el usuario común?

Does the information appear in a logical way for the common user?

2.2 El diseño de los iconos se correspone con objetos cotidianos?

Does the design of the icons correspond to everyday objects?

2.3 Cada icono realiza la acción que el usuario espera?

Each icon performs the action that the user expects?

2.4 Se utilizan frases y conceptos familiares para el usuario?

Are the phrases and concepts used familiar to the user?

3 3 - Control y libertad del usuario / User control and freedom

3.1 Existe un vínculo para volver al estado inicial o a la página de inicio?

Is there a link to return to the initial state or the home page?

3.2 Existen funcionalidades para "deshacer" y "rehacer"?

Are the functions “undo” and “redo” implemented?

3.3 Es fácil volver a un estado anterior de la aplicación?

Is it easy to return to a previous state of the application?

4 4 - Consistencia y estándares / Consistency and standards

4.1 Las etiquetas de los vínculos tienen los mismos nombres que sus destinos?

Do the link labels have the same names as their destinations?

4.2 Las mismas acciones siempre conducen a los mismos resultados?

Ejemplo: Si nos encontramos en un sitio web en el cual, en una interfaz encontramos dos links con el nombre de una misma sección, cada uno de ellos debería llevar al usuario a la misma interfaz, correspondiente a esa sección.

Do the same actions always lead to the same results?

Example: If we are on a website in which, in an interface, we find two links with the name of the same section, each of them should take the user to the same interface, corresponding to that section.

4.3 Un mismo icono tiene el mismo significado en todo el sistema?

Does the same icon have the same meaning throughout the system?

4.4 La información se muestra de forma consistente en todo el sistema?

Is the information consistently displayed throughout the system?

4.5

Los colores de los enlaces son adecuados para su uso?

Are the link colors suitable for use?

4.6 Los elementos de navegación siguen los estándares? (Botones, checkbox, ...)

The navigation elements follow the standards? (Buttons, checkbox, ...)

5 "5 - Reconocimiento en lugar de memoria, aprendizaje y anticipación /

Recognition rather than memory, learning and anticipation"

5.1 Es sencillo utilizar el sistema por vez primera?

Is it easy to use the system for the first time?

5.2 Es fácil localizar información que ya ha sido buscada con anterioridad?

Is it easy to locate information that has already been searched before?

5.3 En todo momento puedes utilizar el sistema sin necesidad de recordar pantallas anteriores?

Can you use the system at any time without having to remember previous screens?

5.4 Todo el contenido necesario para la navegación o para las diferentes tareas está en la "pantalla actual"?

Is all the content needed for navigation or for the different tasks on the “current screen”?

5.5 La información está organizada según la lógica de los usuarios a los que va dirigida la interfície?

Is the information organized according to the logic of the users to whom the interface is directed?

6 6 - Flexibilidad y eficiéncia de uso / Flexibility and efficiency of use

6.1 Existen atajos de teclado para las acciones frecuentes?

Are there keyboard shortcuts for frequent actions?

6.2 Si existen, ¿queda claro cómo usarlas?

If they exist, is it clear how to use them?

6.3 Al realizar cualquier acción por primera vez, queda claro cómo realizarla permanentemente?

When performing any action for the first time, is it clear how to perform it permanently?

6.4 El diseño se adapta al cambiar la resolución de la pantalla?

Does the design adapt when changing the screen resolution?

6.5 Es visible el uso de aceleradores para el usuario habitual?

Is the use of accelerators visible to the regular user?

6.6 Se mantiene siempre ocupado al usuario? (Sin tiempos de espera innecesarios)

Is the user always busy? (Without unnecessary waiting times)

7 "7 - Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errors

Help users recognize, diagnose and recover from errors"

7.1 Se muestra un mensaje antes de tomar acciones irreversibles?

Is a message displayed before taking irreversible actions?

7.2 Los errores cometidos se muestran en tiempo real?

Are the errors shown in real time?

7.3 El mensaje de error que aparece es fácilmente interpretable?

Is the error message that appears easily interpretable?

7.4 Se usa, además, algún código para referenciar el error?

Is some code also used to reference the error?

8 8 - Prevención de errores / Error prevention

8.1 Aparece un mensaje de confirmación antes de realizar las acciones?

Does a confirmation message appear before performing the actions?

8.2 Queda claro qué hay que introducir en cada campo de un formulario?

Is it clear what must be entered in each field of a form?

8.3 El motor de búsqueda tolera errores tipográficos y ortográficos?

Does the search engine tolerate typos and spelling errors?

9 9 - Diseño estético y minimalista / Aesthetic and minimalist design

9.1 Se ha usado un diseño sin redundancia de información?

Has a design without information redundancy been used?

9.2 La información textual es concisa y precisa?

Is the textual information concise and accurate?

9.3 Cada elemento de información se diferencia del resto y es inconfundible?

Each item of information differs from the rest and is unmistakable?

9.4 La estructuración del texto es correcta, con frases cortas i fácil de interpretar?

The structuring of the text is correct, with short sentences and easy to interpret?

10 10 - Ayuda y documentación / Help and documentation

10.1 Existe la opción "ayuda"?

Is there a "help" option?

10.2 En el caso de existir, es visible y de fácil acceso?

If it exists, is it visible and easily accessible?

10.3 La ayuda está orientada a la solución de problemas?

Is the help section aimed at solving problems?

10.4 Dispone de un apartado de preguntas frecuentes?

Is there a section of frequently asked questions (FAQ)?

10.5 La documentación de ayuda es clara, utiliza ejemplos?

Is the help documentation clear, uses examples?

11 11 - Guardar el estado y proteger el trabajo / Save the state and protect the work

11.1 Los usuarios pueden continuar desde un estado anterior al que quedaron en otro momento o desde otro dispositivo?

Can users continue from a previous state where they had previously been or from another device?

11.2 Se implementa la utilidad de "autoguardado"?

Is the "auto-save" utility implemented?

11.3 Tiene buena respuesta a fallos ajenos? (Cortes de corriente, de internet, …)

Does the system have a good response to external failures? (Power cut, internet not working, ...)

12 12 - Color y legibilidad / Colour and readability

12.1 Las fuentes del texto tienen un tamaño adecuado?

Do the text fonts have an appropriate size?

12.2 Las fuentes del texto utilizan colores con suficiente contraste con el fondo?

Do the text fonts use colors with sufficient contrast with the background?

12.3 El texto con imágenes o patrones de fondo es fácil de leer?

Is the text with background images or patterns easy to read?

12.4 Se tiene en cuenta a los usuarios con visión reducida?

Are users with low vision taken into account?

Web Disability Simulator

13 13 - Autonomía / Autonomy

13.1 Se mantiene en todo momento informado al usuario del estado del sistema?

Does it keep the user informed of the system status?

13.2 Además, el estado del sistema es visible y actualitzado?

Moreover, is the system status visible and updated?

13.3 El usuario puede personalizar su espacio personal o de trabajo del sistema?

Can the user customize his personal or work space of the system?

14 14 - Valores per defecto / Default values

14.1 El sistema o aparato proporciona la opción de volver a los valores de fábrica?

Does the system or device provides the option to return to factory settings?

14.2 Si es así, se indican claramente las consecuencias de dicha acción?

If so, are the consequences of such action clearly stated?

14.3 Se utiliza el término “por defecto”?

Is the term "default" used?

15 15 - Reducción de la latencia / Latency reduction

15.1 La ejecución de tareas pesadas es transparente al usuario?

Is the execution of heavy work transparent to the user?

15.2 Se muestra el tiempo restante o alguna animación de las tareas pesadas que se están ejecutando?

While heavy tasks are being executed, is it showing the remaining time or some animation?

Usability website sources: Nielsen: https://www.nngroup.com/articles/ten-usability-heuristics/

Structure:

Generate HTML, CSS, and JS code with accessible tags that validate the HTML, CSS, and JS code and comply with usability and accessibility aspects.

Design

Use the design system: https://carbondesignsystem.com/

Design colour: Include as primary colour: #1F70B8. Consider the contrast of all elements.

Legible sans-serif typography (e.g. Arial, Verdana, Open Sans).

Large buttons and sufficient separation between interactive elements.

Includes responsive features such as content organisation when the screen is small.

Change menu items to a mobile-friendly menu.

Interaction:

Include interactions with dynamic information that appears on the page.

Include content sections that are displayed dynamically.

Ensure that interactions are keyboard accessible and compatible with screen readers.

Include WAI-ARIA if necessary.

EXPECTED OUTPUT:

Submit the complete code in HTML, CSS, and JS in a single block or structured in three parts. Do not use external frameworks (such as React, Angular, etc.). Do not use external JS libraries.

FINAL INSTRUCTIONS:

Do you require further information about the course, the target audience, or the content structure to complete this task?

**Step 4. Prompt improvement**

OPTIMIZED PROMT (after manual creation, review by chat gpt, and promt optimization tool)

Role

You are a senior web designer specializing in accessibility and usability.

Objective

Generate the complete code (HTML, CSS, and JavaScript) for a landing page for an accessibility course, meeting the highest standards of web accessibility.

1. Accessibility requirements

1. WCAG 2.2: cover all level A and AA criteria.

2. Cognitive inclusion: apply guidelines from “Making Content Usable for People with Cognitive and Learning Disabilities.”

3. WAI ARIA: label roles, states, and properties where appropriate.

4. Clear language: easy-to-read text, avoid technical jargon.

5. Keyboard navigation: full support, logical tab order, and landmarks.

2. Usability heuristics

Apply these 15 heuristics (first level), with their numbering in code comments:

1. System visibility and status

2. Connection to the real world

3. User control and freedom

4. Consistency and standards

5. Recognition vs. memory

6. Flexibility and efficiency

7. Error diagnosis and recovery

8. Error prevention

9. Aesthetic and minimalist design

10. Help and documentation

11. State saving and protection

12. Color and readability

13. Autonomy

14. Default values

15. Latency reduction

3. Visual design

• System: Carbon Design System (https://carbondesignsystem.com/)

• Palette:

o Primary: #1F70B8

o Secondary: choose accent color with WCAG contrast ≥ 4.5:1

• Typography: sans serif (e.g., Open Sans), sizes defined in rem.

• Spacing: generous margins and paddings (≥ 1rem).

• Responsive:

o Desktop, tablet, and mobile (breakpoints: 1024px, 768px, 480px).

o Collapsible menu (hamburger) at < 768px.

4. Behavior and interactions

• Collapsible sections: keyboard controllable and ARIA announcement when expanding/collapsing.

• Forms: real-time validation, clear messages, and aria-live.

• Animations: smooth transitions (< 300 ms), no flickering (> 3 flashes).

5. Code delivery

HTML:

o Semantic and validated (W3C).

o Comments indicating section and heuristics applied.

CSS (vanilla):

o Modular files, CSS variables for colors/spacing.

o BEM classes or similar for organization.

JavaScript (pure):

o ES6 modules, no external dependencies.

o Documented accessibility events (focus, keydown).

Structure: three root folders: /html, /css, /js.

JSON FILE WITH CONTENT (Direct JSON file)

Next step

When you're ready, send me the JSON with the content structure (sections, titles, texts, labels, CTAs, FAQs, etc.) so I can inject it into the template.

**Step 5. Prompt customization.**

## Variaciones de promts

### vercel

CON ESTE CONTENIDO (ANTERIOR), CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

Vercel v0 (<https://v0.dev/>)

# Vercel v0 Prompt for Landing Page Generator

## 🧑‍💼 Role

You are a \*\*Senior Web Designer\*\* specializing in \*\*Accessibility\*\* and \*\*Usability\*\*.

## 🎯 Objective

Generate the \*\*complete code\*\* (HTML, CSS, JavaScript) for a \*\*landing page\*\* promoting an accessibility course. The output must be a ZIP-like project structure ready to deploy on Vercel v0.

---

## 1. Accessibility Requirements

1. \*\*WCAG 2.2\*\*: satisfy \*\*ALL\*\* Level A & AA success criteria.

2. \*\*Cognitive Inclusion\*\*: implement “Making Content Usable for People with Cognitive and Learning Disabilities.”

3. \*\*WAI‑ARIA\*\*: add appropriate `role`, `aria-\*` attributes, and states.

4. \*\*Clear Language\*\*: use easy‑read text; avoid jargon.

5. \*\*Keyboard Navigation\*\*: full support, logical `tabindex`, and semantic landmarks (`<header>`, `<main>`, `<nav>`, `<footer>`).

---

## 2. Usability Heuristics

1. Visibility & system status

2. Real‑world connection & metaphors

3. User control & freedom

4. Consistency & standards

5. Recognition vs. memory

6. Flexibility & efficiency

7. Error diagnosis & recovery

8. Error prevention

9. Aesthetic & minimalist design

10. Help & documentation

11. Save state & work protection

12. Color & readability

13. Autonomy

14. Default values

15. Latency reduction

---

## 3. Visual Design

- \*\*Design System\*\*: Carbon Design System (https://carbondesignsystem.com/)

- \*\*Color Palette\*\*

- Primary: `#1F70B8`

- Accent: choose one with \*\*WCAG ≥ 4.5:1\*\* contrast.

- \*\*Typography\*\*: sans‑serif (e.g. Open Sans), sizes in `rem`.

- \*\*Spacing\*\*: margin/padding ≥ `1rem`.

- \*\*Responsive Breakpoints\*\*

- Desktop: ≥ 1024 px

- Tablet: 768–1023 px

- Mobile: < 768 px (hamburger menu)

---

## 4. Interaction & Behavior

- \*\*Collapsible Sections\*\*: keyboard‑toggleable, announce state changes with `aria-expanded`.

- \*\*Forms\*\*: real‑time validation, clear error messages, `aria-live` regions.

---

## 5. Code Delivery

- \*\*Project Structure\*\* (root):

/html  
index.html  
/css  
variables.css  
main.css  
/js  
ui-interactions.js

- \*\*HTML\*\*

- Semantic, W3C‑valid.

- Comments marking \*\*sections\*\* and \*\*heuristic tags\*\*.

- \*\*CSS\*\* (vanilla)

- Modular files, custom properties (`--color-primary`, `--spacing-base`).

- BEM or similar naming.

- \*\*JavaScript\*\* (plain ES6 modules)

- No external dependencies.

- Document focus, keydown handlers, and ARIA updates.

---

DOCUMENT JSON WITH THE CONTENT

(fichero json)

### Builder.io

CON ESTE CONTENIDO --- CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

**Builder.io**

Create a responsive, accessible landing page in Spanish for a university microcredential program.

Use semantic HTML and Tailwind CSS. Apply WCAG 2.2 accessibility practices and include ARIA where needed. Follow logical heading order, keyboard navigation, and contrast compliance.

Structure the page as follows:

1. <html lang="es"> and set the page title as "Microcredencial en Accesibilidad Digital"

2. Header:

- Include a logo (logo.jpg)

- Course title as an <h1>: “Microcredencial Universitaria en Especialización en Accesibilidad Digital por la Universitat de Lleida”

3. Navigation:

- Horizontal navigation bar with links: Presentación, Profesorado, Programa, Matrícula, FAQ

4. Main section:

- Intro paragraph describing the course purpose and audience

- Three course features in cards or columns:

- Reconocimiento oficial

- Aprendizaje práctico con casos reales

- Profesorado experto en accesibilidad

- Call to action button: “Inscribirse ahora”

5. Accessibility best practices:

- Use alt texts for images

- Ensure headings are sequential (h1 > h2 > h3…)

- Ensure contrast ratios comply with 1.4.3 AA

- Use focus outlines and keyboard navigation (2.4.7 AA)

- Add hover/focus feedback for links (1.4.13\*)

6. Footer:

- Contact email and phone

- Logos of Universitat de Lleida

- Links: política de privacidad, política de accesibilidad

- Set role="contentinfo" on the footer

Design: clean, modern, and professional. Use accessible color contrast and readable typography.

DOCUMENT JSON WITH THE CONTENT

(fichero json)

### **DEEPSITE.AI**

CON ESTE CONTENIDO --- CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

### 🔧 OPTIMIZED PROMPT FOR **DEEPSITE.AI**

**Role:**  
You are a senior web designer specialized in accessibility and usability.

**Goal:**  
Generate a complete landing page (HTML, CSS, JavaScript) for a **Digital Accessibility Training Course**, following the **highest accessibility and usability standards**, using modern, semantic, and modular front-end practices.

### ✅ Accessibility Requirements

Ensure **full compliance** with:

1. **WCAG 2.2 (Level A and AA)** – all success criteria must be addressed.
2. **Cognitive accessibility** based on “Making Content Usable for People with Cognitive and Learning Disabilities”.
3. **WAI-ARIA**: add relevant roles, states, and properties for screen reader support.
4. **Plain language**: use clear, simple text with no technical jargon.
5. **Keyboard navigation**: logical tab order, all elements reachable by keyboard, with semantic landmarks (<main>, <nav>, <header>, etc.).

### 💡 Usability Heuristics (annotate in code comments)

Implement the following 15 usability heuristics, referencing each with numbered comments:

1. System status visibility
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Recognition rather than recall
6. Flexibility and efficiency
7. Error diagnosis and recovery
8. Error prevention
9. Aesthetic and minimalist design
10. Help and documentation
11. Save state and user safety
12. Color contrast and legibility
13. User autonomy
14. Smart defaults
15. Low latency

### 🎨 Visual Design Guidelines

* **Design System**: [Carbon Design System](https://carbondesignsystem.com/)
* **Color palette**:
  + Primary: #1F70B8
  + Secondary: custom accent color with **contrast ratio ≥ 4.5:1 (WCAG compliant)**
* **Typography**: Open Sans or similar sans-serif, sizes in rem
* **Spacing**: padding/margin ≥ 1rem
* **Responsive breakpoints**:
  + >1024px (desktop)
  + 768px–1024px (tablet)
  + <768px (mobile, hamburger menu enabled)

### 🧩 Interactions & Behavior

* **Collapsible sections**:
  + Fully keyboard-accessible
  + Use aria-expanded, aria-controls, and aria-live as appropriate
* **Form validation**:
  + Real-time, accessible feedback with clear instructions
  + Use aria-invalid, aria-describedby, aria-live
* **Animations**:
  + Transitions must be under 300ms
  + Avoid flashing content (>3 flashes/sec)

### 📁 Output Structure

Generate a working project with the following root folders:

* /html: semantic HTML5 with **W3C validation**
* /css: vanilla CSS, modular, with CSS variables (colors, spacing) and BEM-style class names
* /js: native JavaScript (ES6 modules), no external dependencies, accessibility-focused events documented

DOCUMENT JSON WITH THE CONTENT

(fichero json)

### BOLT

<https://bolt.new/>

CON ESTE CONTENIDO --- CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

**⚡ OPTIMIZED PROMPT FOR** [**BOLT.NEW**](https://bolt.new/)

**Role:**  
You are a senior web designer specializing in **web accessibility** and **usability**.

**Goal:**  
Generate a complete and accessible **landing page** for a digital accessibility course. The page must follow **WCAG 2.2 (Level A and AA)**, support **cognitive accessibility**, follow **WAI-ARIA best practices**, and apply **usability heuristics** within modern responsive design.

**✅ Accessibility Requirements**

Implement:

1. **WCAG 2.2 (Level A & AA)**: all success criteria must be respected
2. **Cognitive accessibility**: apply *Making Content Usable for People with Cognitive and Learning Disabilities*
3. **WAI-ARIA** roles, states, and properties where appropriate
4. **Plain language**: write in clear, easy-to-read language with no technical jargon
5. **Full keyboard support** with logical tab order and semantic landmarks (<header>, <nav>, <main>, etc.)

**💡 Usability Heuristics**

Apply the following **15 usability heuristics** and annotate them in code comments using numbered references (e.g. // Heuristic 3: User control and freedom):

1. System status visibility
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Recognition rather than recall
6. Flexibility and efficiency
7. Error diagnosis and recovery
8. Error prevention
9. Aesthetic and minimalist design
10. Help and documentation
11. Save state and user protection
12. Color and readability
13. User autonomy
14. Useful defaults
15. Low latency interaction

**🎨 Visual Design**

* **Design system**: [Carbon Design System](https://carbondesignsystem.com/)
* **Colors**:
  + Primary: #1F70B8
  + Secondary: any accent color with contrast ratio ≥ 4.5:1 (WCAG compliant)
* **Typography**: Open Sans or similar sans-serif font, sizes in rem
* **Spacing**: generous spacing (≥ 1rem)
* **Responsive design**:
  + Breakpoints: 1024px, 768px, 480px
  + Collapsible hamburger menu below 768px

**🧩 Interactions & Behavior**

* **Collapsible sections**: keyboard operable, aria-expanded, aria-controls
* **Forms**:
  + Real-time validation with aria-live announcements
  + Clear labeling and error messages
* **Animations**:
  + Smooth transitions under 300ms
  + No flashing or strobe effects (avoid >3 flashes/sec)

**📁 Code Delivery Guidelines**

**Project structure (three folders):**

* /html: semantic HTML, W3C validated, with comments for heuristics
* /css: vanilla modular CSS with BEM-style class names and CSS variables
* /js: pure ES6 JavaScript, modular, no external dependencies, with documented accessibility event handlers (focus, keydown, etc.)

DOCUMENT JSON WITH THE CONTENT

(fichero json)

ChatGPT

CON ESTE CONTENIDO --- CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

**PROMPT PARA CHATGPT**

Eres un diseñador web senior, especialista en accesibilidad y usabilidad. Tu tarea es **generar el código completo** (HTML, CSS y JavaScript puro) para una landing page de un curso de accesibilidad, cumpliendo los más altos estándares técnicos, de diseño y de accesibilidad.

**🎯 Objetivo**

Crear una landing page para el curso **"Microcredencial en Especialización en Accesibilidad Digital"** ofrecido por la Universitat de Lleida.

**✅ Criterios de accesibilidad (obligatorios)**

1. **WCAG 2.2** (Nivel A y AA)
2. **Inclusión cognitiva** (según “Making Content Usable for People with Cognitive and Learning Disabilities”)
3. **WAI-ARIA**: roles y estados bien aplicados.
4. **Lenguaje claro** y lectura fácil.
5. **Navegación por teclado** y landmarks bien definidos.

**🎨 Diseño visual**

* Sistema: **Carbon Design System**
* Colores:
  + Primario: #1F70B8
  + Secundario: color de acento con ratio de contraste WCAG ≥ 4.5:1
* Tipografía: **Open Sans**, tamaños en rem
* Responsive: breakpoints a 1024px, 768px, 480px
* Menú hamburguesa en pantallas pequeñas

**⚙️ Comportamiento e interacciones**

* Secciones colapsables con accesibilidad
* Formularios con validación accesible (aria-live)
* Transiciones suaves (< 300 ms), sin parpadeos

**📁 Entrega de código**

* HTML semántico con comentarios que indiquen heurística aplicada (1-15)
* CSS vanilla, con BEM, variables CSS
* JS en ES6, sin librerías, accesible y documentado
* Estructura:

bash

CopiarEditar

/html

/css

/js

**📦 Estructura base del contenido (JSON)**

Utiliza esta estructura en JSON como fuente de contenido para generar el sitio web:

Fichero JSON

Segundo. Refinado (para chat gpt)

**🧠 PROMPT ÓPTIMO PARA CHATGPT**

**Objetivo**: Generar una landing page accesible a partir de un JSON estructurado  
**Plataforma**: ChatGPT  
**Rol asignado**: Diseñador/a web senior especializado/a en accesibilidad y usabilidad

**🎯 TU TAREA**

Genera el **código completo** de una landing page (HTML, CSS y JavaScript puro) para un curso de accesibilidad digital, utilizando el contenido que se te proporciona en formato JSON.

Debes cumplir con las siguientes **especificaciones** de accesibilidad, diseño visual, comportamiento e interacción.

**📚 1. CONTENIDO BASE (JSON)**

Usa este JSON como contenido para poblar la estructura del sitio web:  
🔹 Navegación, módulos, formularios, profesorado, etc.  
🔹 Es importante respetar los elementos clave y sus IDs para la navegación por anclajes.

*(Aquí se incluirá el JSON estructurado, como el que ya hemos generado en la última iteración.)*

**✅ 2. CRITERIOS DE ACCESIBILIDAD (OBLIGATORIOS)**

Cumple los siguientes requisitos al generar el código:

1. **WCAG 2.2 (A y AA)**: asegúrate de cumplir todos los criterios actuales.
2. **Inclusión cognitiva**: aplica pautas de *“Making Content Usable for People with Cognitive and Learning Disabilities”*.
3. **WAI-ARIA**: implementa roles, propiedades y estados donde sea necesario.
4. **Lenguaje claro**: textos en lectura fácil, sin tecnicismos innecesarios.
5. **Navegación por teclado**: soporte completo, landmarks y orden lógico de tabulación.

**🎨 3. DISEÑO VISUAL**

* **Sistema de diseño**: [Carbon Design System](https://carbondesignsystem.com/)
* **Colores**:
  + Primario: #1F70B8
  + Secundario: color con ratio de contraste WCAG ≥ 4.5:1
* **Tipografía**: Sans-serif (p.ej. Open Sans), en unidades rem
* **Responsive**:
  + Breakpoints: 1024px, 768px, 480px
  + Menú hamburguesa en móviles (<768px)
* **Espaciado**: márgenes y paddings generosos (≥ 1rem)

**⚙️ 4. COMPORTAMIENTO E INTERACCIONES**

* **Secciones colapsables**: accesibles por teclado, con aria-expanded y aria-controls
* **Formularios**: validación en tiempo real con mensajes accesibles (aria-live)
* **Animaciones**: transiciones suaves (< 300 ms), sin parpadeos ni flashes (máximo 3)

**🧩 5. HEURÍSTICAS DE USABILIDAD**

Añade en comentarios de código referencias a estas heurísticas (usando su número):

1. Visibilidad del estado del sistema
2. Conexión con el mundo real
3. Control y libertad del usuario
4. Consistencia y estándares
5. Reconocimiento en lugar de recuerdo
6. Flexibilidad y eficiencia
7. Diagnóstico y recuperación de errores
8. Prevención de errores
9. Diseño estético y minimalista
10. Ayuda y documentación
11. Guardar estado y protección de datos
12. Color y legibilidad
13. Autonomía
14. Valores por defecto
15. Reducción de la latencia

**📁 6. ENTREGA DE CÓDIGO (ESTRUCTURA)**

Organiza el proyecto así:

sql

CopiarEditar

/html --> index.html (semántico y validado W3C)

/css --> estilos.css (vanilla CSS con variables y BEM)

/js --> main.js (ES6, sin dependencias, accesible)

**⚠️ INSTRUCCIONES FINALES**

* El código debe ser accesible, semántico, limpio y comentado.
* No uses frameworks ni librerías externas.
* No olvides incluir landmarks (<main>, <nav>, <header>, <footer>, etc.).
* Genera todo en una sola respuesta si es posible, o por partes bien etiquetadas si es muy largo.

### Claude

CON ESTE CONTENIDO --- CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

Aquí tienes un **prompt optimizado para Claude** (Anthropic) para generar una **landing page accesible para un curso**, usando el contenido y requerimientos que has proporcionado, incluyendo el sistema de diseño **Carbon Design System** y el contenido JSON estructurado que has mencionado anteriormente:

### 🧠 **Prompt para Claude**

Eres un diseñador web senior especializado en **accesibilidad digital** y **usabilidad**. Tu objetivo es generar el **código completo (HTML, CSS y JavaScript)** de una **landing page para un curso universitario de accesibilidad digital**, siguiendo los más altos estándares de accesibilidad y diseño profesional.

Utiliza este conjunto de directrices:

#### 🎯 OBJETIVO

Desarrollar una landing page informativa y accesible para una **microcredencial universitaria en accesibilidad digital**. Debe transmitir confianza, ser clara, legible y fácil de navegar.

### 🔐 REQUISITOS DE ACCESIBILIDAD

1. **WCAG 2.2**: cubrir todos los criterios de nivel A y AA.
2. **Cognición**: aplicar pautas de Making Content Usable for People with Cognitive and Learning Disabilities.
3. **WAI-ARIA**: usar roles, propiedades y estados para mejorar la comprensión y navegación por asistentes.
4. **Lenguaje claro**: estilo de lectura fácil, evitar tecnicismos innecesarios.
5. **Navegación por teclado**: orden lógico de tabulación, sin trampas, con puntos de referencia semánticos (<header>, <main>, <nav>, <footer>, etc.).

### 👁️ HEURÍSTICAS DE USABILIDAD A INCLUIR (como comentarios en el HTML):

1. Visibilidad del estado del sistema
2. Conexión con el mundo real
3. Control del usuario
4. Consistencia
5. Reconocimiento y no memoria
6. Eficiencia y flexibilidad
7. Diagnóstico y recuperación de errores
8. Prevención de errores
9. Estética y minimalismo
10. Ayuda contextual
11. Guardado y protección de estado
12. Legibilidad de color
13. Autonomía del usuario
14. Valores por defecto
15. Reducción de la latencia

### 🎨 DISEÑO VISUAL

* **Sistema visual**: usa [Carbon Design System](https://carbondesignsystem.com/), complementado con una hoja de estilos personalizada (modular).
* **Paleta**:
  + Primario: #1F70B8
  + Secundario: alto contraste con mínimo WCAG 4.5:1
* **Tipografía**: sans-serif moderna ("IBM Plex Sans", "Open Sans")
* **Espaciado**: mínimo 1rem, usar márgenes generosos.
* **Responsive**: breakpoints en 1024px, 768px, 480px. Menú hamburguesa para < 768px.

### 🤖 COMPORTAMIENTO / INTERACCIONES

* Secciones colapsables accesibles por teclado (<details> o ARIA aria-expanded)
* Formularios con validación accesible en tiempo real (ARIA live, mensajes claros)
* Transiciones suaves (< 300 ms), sin parpadeos

### 📁 ESTRUCTURA DEL CÓDIGO

Genera tres carpetas raíz:

bash

CopiarEditar

/html → index.html completo, con comentarios de heurísticas

/css → styles.css personalizada compatible con Carbon

/js → main.js con interacciones accesibles

### 📦 CONTENIDO DE LA LANDING PAGE (estructura propuesta)

1. **Presentación del curso**
2. **Programa académico**
3. **Profesorado**
4. **Salidas profesionales**
5. **Requisitos de acceso**
6. **Matrícula y precio**
7. **Formulario de contacto accesible**

### 🧾 OUTPUT ESPERADO

* Código **HTML semántico y validado**, con comentarios explicando cada heurística aplicada.
* **CSS modular vanilla** con variables CSS (:root) para colores, espaciado, y uso de clases BEM.
* **JavaScript puro** ES6, sin frameworks. Enfocado en eventos accesibles (focus, keydown, aria-\*, etc.)
* Todo el código debe ser **copiable y funcional directamente** (sin dependencias externas).

### Copilot

CON ESTE CONTENIDO --- CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

Aquí tienes un **prompt optimizado para Copilot (GitHub Copilot)**, teniendo en cuenta que Copilot responde mejor a instrucciones **contextuales en comentarios dentro del código**, siguiendo el estilo de doc-comment driven development y ofreciendo completions predictivas a partir de código inicial.

El enfoque consiste en definir claramente el propósito en comentarios estructurados, para que Copilot entienda el objetivo y genere completaciones útiles, especialmente al trabajar dentro de un entorno de desarrollo como VS Code.

### ✅ PROMPT PARA COPILOT (archivo README.md o bloque de comentario inicial)

/\*\*  
👨‍💻 Proyecto: Landing Page Accesible para Curso de Accesibilidad Digital  
🎯 Objetivo: Generar el código completo (HTML, CSS y JavaScript) para una landing page cumpliendo **los más altos estándares de accesibilidad y usabilidad**.  
📦 Estructura: /html, /css, /js, y un JSON de contenido.

♿ **1. Accesibilidad obligatoria**

* Cumplir todas las WCAG 2.2 (nivel A y AA)
* Aplicar pautas cognitivas (Making Content Usable for People with Cognitive and Learning Disabilities)
* WAI-ARIA: roles, propiedades y estados donde apliquen
* Lenguaje claro y lectura fácil
* Navegación por teclado completa, con landmarks semánticos

🧠 **2. Heurísticas de usabilidad** (añadir en comentarios del HTML):

1. Visibilidad del estado del sistema
2. Conexión con el mundo real
3. Control del usuario
4. Consistencia
5. Reconocimiento vs. memoria
6. Flexibilidad
7. Diagnóstico y recuperación
8. Prevención de errores
9. Estética y minimalismo
10. Ayuda contextual
11. Guardado de estado
12. Legibilidad de color
13. Autonomía
14. Valores por defecto
15. Reducción de latencia

🎨 **3. Diseño visual**

* Basado en [Carbon Design System](https://carbondesignsystem.com)
* Paleta:
  + Primario: #1F70B8
  + Secundario: contraste ≥ 4.5:1
* Tipografía: "Open Sans", tamaños en rem
* Responsive: 1024px / 768px / 480px (menú hamburguesa < 768px)

🤖 **4. Comportamiento e interacciones**

* Secciones colapsables accesibles con teclado (aria-expanded, aria-controls)
* Formularios con validación en tiempo real y aria-live
* Transiciones suaves < 300 ms
* Sin parpadeos o animaciones intensas (> 3 flashes)

📁 **5. Entrega estructurada**

* /html/index.html con comentarios de heurísticas
* /css/styles.css modular, con variables en :root
* /js/main.js en ES6 sin frameworks
* contenido.json con datos de la landing page

Copilot, por favor sugiere:

* HTML semántico y accesible con comentarios de heurísticas
* CSS vanilla compatible con Carbon Design System
* JS accesible para interacciones (detalles, formularios, foco)
* Lectura del fichero JSON si se desea contenido dinámico  
  \*/

### 💡 Consejos de uso en el entorno Copilot

* Abre index.html, y empieza escribiendo:

html

CopiarEditar

<!-- Sección 1: Header (heurística 1, 2, 4) -->

<header class="bx--header" role="banner">

Copilot completará con código contextual respetando el diseño Carbon + accesibilidad.

* En el CSS, empieza con:

css

CopiarEditar

/\* :root variables para colores y fuentes según Carbon Design System \*/

:root {

* En JS:

js

CopiarEditar

// Módulo para control accesible de secciones colapsables (heurísticas 3, 6, 8)

document.querySelectorAll('[aria-expanded]').forEach((toggle) => {

### Gemini

CON ESTE CONTENIDO --- CREA UN PROMT OPTIMIZADO PARA LA PLATAFORMA:

Aquí tienes un **prompt optimizado para Google Gemini** (o Gemini Code Assist) basado en tus requisitos de accesibilidad, usabilidad y diseño visual. Está estructurado para facilitar la generación coherente de HTML, CSS y JS a partir del fichero JSON, con buenas prácticas de desarrollo y comentarios para IA generativas avanzadas:

Eres un desarrollador web especializado en accesibilidad y usabilidad. Utiliza los siguientes requisitos para generar el \*\*código completo y final de una landing page accesible\*\*, siguiendo los estándares web más exigentes. La estructura debe estar dividida en tres carpetas: `/html`, `/css`, `/js`.

🧩 CONTEXTO GENERAL:

- Curso: Microcredencial Universitaria en Accesibilidad Digital.

- Objetivo de la página: promocionar el curso e informar de forma accesible.

- Fuente de contenido: fichero JSON que define todas las secciones del curso (Presentación, Programa académico, Profesorado, Salidas profesionales, Requisitos de acceso, Matrícula y precio).

🔐 ACCESIBILIDAD (cumplimiento obligatorio):

1. WCAG 2.2: todos los criterios A y AA.

2. Inclusión cognitiva: aplicar pautas de “Making Content Usable for People with Cognitive and Learning Disabilities”.

3. WAI-ARIA: roles, propiedades y estados correctamente etiquetados.

4. Lenguaje claro: sin tecnicismos, lectura fácil.

5. Navegación por teclado: tabulación lógica, landmarks y foco visible.

🧠 USABILIDAD:

Aplica estas \*\*15 heurísticas\*\* de usabilidad, comentando en el código con su número correspondiente. Ejemplo: `<!-- Heurística 2: Conexión con el mundo real -->`.

🎨 DISEÑO VISUAL:

- Sistema: Carbon Design System (https://carbondesignsystem.com/)

- Paleta: primario `#1F70B8`, secundario con contraste ≥ 4.5:1.

- Tipografía: sans serif (p.ej. Open Sans), tamaños en `rem`.

- Espaciado mínimo: `1rem`.

- Responsive: breakpoints 1024px, 768px, 480px.

- Menú colapsable tipo hamburguesa en pantallas < 768px.

🧩 INTERACCIONES:

- Secciones colapsables (teclado + `aria-expanded`, `aria-controls`).

- Formularios con validación en tiempo real, mensajes accesibles (`aria-live`).

- Transiciones suaves (< 300ms), sin flashes (> 3 por segundo).

📁 ESTRUCTURA DE FICHEROS:

- `/html/index.html`: estructura semántica completa, con landmarks (`<main>`, `<nav>`, `<header>`, etc.).

- `/css/styles.css`: vanilla CSS, BEM o similar, variables CSS para colores, tipografías y spacing.

- `/js/main.js`: ES6 modular, sin frameworks, interacciones accesibles (eventos `keydown`, `focus`, `click`, etc.), comentarios explicativos.

📦 INPUT (estructura JSON del curso):

Incluye aquí el fichero JSON que contiene todo el contenido de la página (si ya lo tienes generado). Usa los campos para generar dinámicamente las secciones de la web con su contenido.

⚙️ OBJETIVO FINAL:

Generar una landing page completamente funcional, validada por W3C, con código limpio y mantenible, siguiendo estándares profesionales.