

DOCUMENTACIÓN CDC

Sergio Pérez
Roberto García
Diego González
Antonio Constantin

ÍNDICE

1	INTERFACES	2
1.1	INTRODUCCIÓN	2
1.2	PROTOTIPOS	3
1.2.1	SKECTCH MÓVIL.....	3
1.2.2	SKETCH ORDENADOR.....	4
1.2.3	SKETCH ORDENADOR SPRINT 2.....	6
1.2.4	SKETCH MOVIL SPRINT 2.....	7
1.3	ESTRUCTURA.....	8
1.4	ESTILO VISUAL.....	10
1.4.1	TIPOGRAFÍA	10
1.4.2	COLORES.....	11
1.5	RESULTADO LANDING PAGE.....	12
2	GITHUB.....	15
3	CLIENTE	18
3.1	Formulario HTML	18
3.2	Validaciones JS.....	19
3.3	Comprobación HTML	20
3.4	Carrito.....	23
3.5	Interfaces.....	26
3.5.1	DI6	26
3.5.2	DI7	27

1 INTERFACES

1.1 INTRODUCCIÓN

Esta documentación tiene como objetivo proporcionar una visión detallada y completa de los prototipos de la landing page, abarcando aspectos cruciales como la estructura, el estilo visual, tipografía y paleta de colores. Además, se incluyen elementos esenciales como el repositorio en GitHub, el formulario del cliente con validaciones detalladas, y los requisitos técnicos que han dado forma a este proyecto.

Estructura de la Página:

- Se detallan los componentes estructurales de la landing page, desde la disposición de los elementos hasta la navegación, asegurando una experiencia de usuario fluida y coherente.
- Se describen los bloques de contenido, la disposición de las secciones y cualquier interactividad planificada.
- Estilo Visual: Esta sección aborda la estética general de la página. Se detallan las decisiones de diseño, la disposición de elementos visuales y la cohesión general del estilo. Se hace especial énfasis en la experiencia del usuario y la accesibilidad visual.
- Tipografía y Paleta de Colores: Se presenta la tipografía seleccionada, justificando las decisiones tomadas en cuanto a legibilidad y estilo. Asimismo, se documenta la paleta de colores empleada, explicando su relación con la identidad de la marca y cómo contribuye a la percepción general de la página.
- Repositorio en GitHub: Se proporciona el enlace al repositorio en GitHub donde se almacenan todos los archivos relacionados con el desarrollo de la landing page. Este repositorio sirve como fuente centralizada y permite un control de versiones eficiente para futuras actualizaciones y colaboraciones.
- Formulario del Cliente: Se analiza en detalle el formulario que el cliente utiliza, incluyendo los campos obligatorios, el tipo de datos esperado, y las validaciones implementadas para garantizar la integridad de la información recibida. Se documentan los mensajes de error que se despliegan en caso de ingreso incorrecto de datos.

En resumen, esta guía de estilo no solo sirve como un compendio informativo, sino también como un recurso vital para el mantenimiento continuo y el desarrollo coherente de la landing page. Proporciona una base sólida para futuras modificaciones y asegura que cualquier contribución al proyecto se realice de manera informada y eficiente.

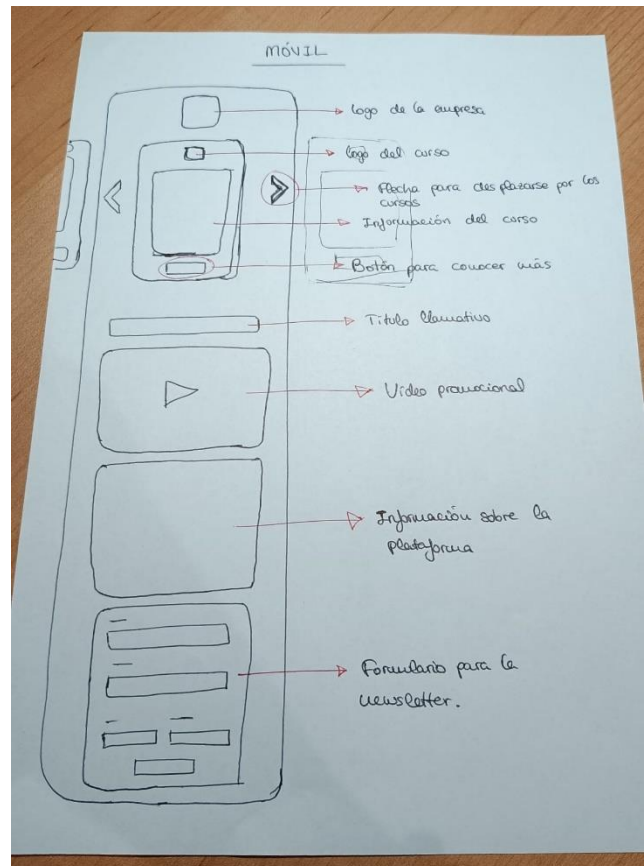
1.2 PROTOTIPOS

1.2.1 SKETCH MÓVIL

Sketch móvil (digital)

The digital prototype shows a mobile app interface. At the top is the CDC logo. Below it is a purple banner with the Java logo and text: "¿QUIERES SER UN EXPERTO EN JAVA? NUESTRO CURSO TE LLEVARÁ DESDE LOS FUNDAMENTOS HASTA LA MAESTRÍA EN ESTE LENGUAJE DE PROGRAMACIÓN CLAVE. ABRE LAS PUERTAS A EMOCIONANTES OPORTUNIDADES EN EL MUNDO DE LA TECNOLOGÍA. ¡INSCRIBETE AHORA Y DESPEGA EN TU CARRERA DE DESARROLLO DE SOFTWARE!". A "GET STARTED" button is at the bottom of the banner. Below the banner is the text "TENEMOS MUCHÍSIMOS CURSOS". This is followed by a "CURSO BÁSICO" banner with the Java logo. Below that is the text: "ADQUIERE LAS HABILIDADES NECESARIAS A TRAVÉS DE UN COMPLETO CURSO DE PROGRAMACIÓN Y APRENDE CODING PARA CONVERTIRTE EN UN DESARROLLADOR FULL STACK." At the bottom is a registration form with fields for "Apellido:", "Logot", "Email:" (with the example "mariolopez@meowback.com"), "Teléfono:", a checkbox for "¡No he leído los términos de uso, ni la privacidad!", and a "Modalidad de estudio:" section with "Online" and "Presencial" options. A "Solicita tu plaza ahora" button is at the bottom, followed by a small disclaimer about data processing.

Sketch móvil (dibujo)



1.2.2 SKETCH ORDENADOR

Sketch móvil (digital)



TENEMOS MUCHÍSIMOS CURSOS



ADQUIERE LAS HABILIDADES NECESARIAS A TRAVÉS DE UN COMPLETO CURSO DE PROGRAMACIÓN Y APRENDE CODING PARA CONVERTIRTE EN UN DESARROLLADOR FULL STACK. PODRÁS CREAR TUS PROPIAS APLICACIONES WEB DESDE CERO Y CONTAR CON EL APOYO NECESARIO PARA COMENZAR EXITOSAMENTE TU CARRERA EN EL SECTOR TECNOLÓGICO.

No te quedes sin tu plaza en el próximo bootcamp.

Elige entre dos modalidades: 9 semanas a tiempo completo o 24 semanas a tiempo parcial, con clases por la tarde después de trabajar.

Nombre: *

Marta

Apellido: *

López

Email: [✉](#)

martalopez@ironhack.com

Teléfono:

¡No habrá llamadas de spam, te lo prometemos!

Modalidad de estudio *

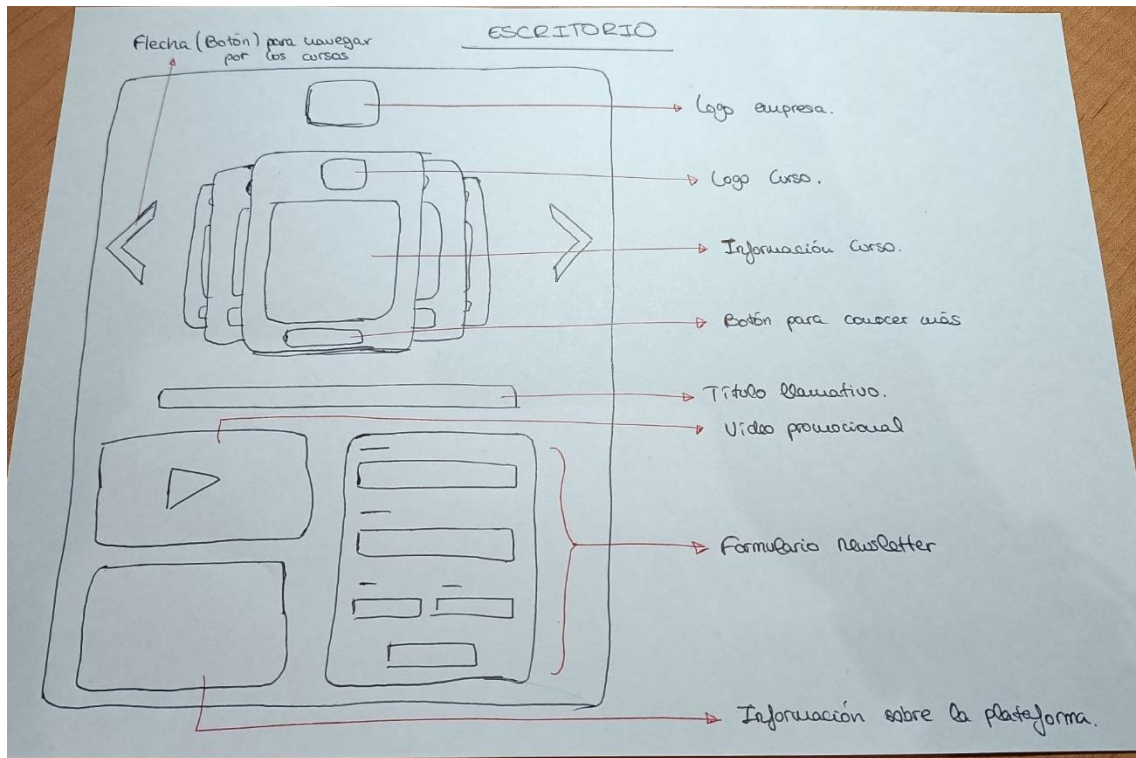
Online

Presencial

Solicita tu plaza ahora

Al enviar el formulario, acepta el procesamiento de sus datos personales de acuerdo con la [Política de privacidad](#) y acepta los [términos y condiciones](#).

Sketch ordenador (dibujo)



1.2.3 SKETCH ORDENADOR SPRINT 2



HTML



CSS



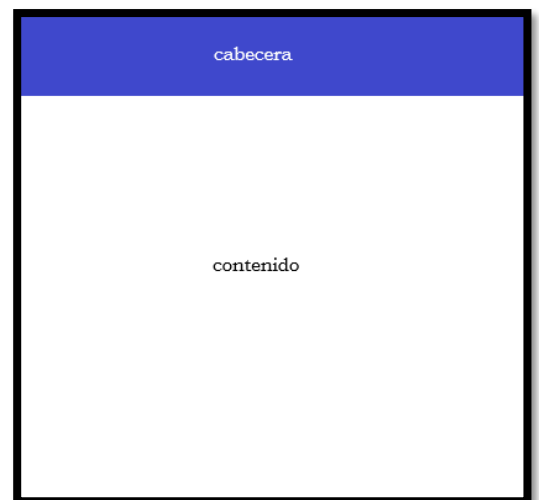
1.2.4 SKETCH MOVIL SPRINT 2



1.3 ESTRUCTURA

A la hora de diseñar la estructura de nuestra página hemos optado por un esquema básico basado en contenedores grandes separados horizontalmente. Esta estructura nos permite tener facilidad en la navegación y flexibilidad para incorporar nuevos contenidos.

En esta estructura encontramos, de arriba hacia abajo, la cabecera y el bloque de contenido:

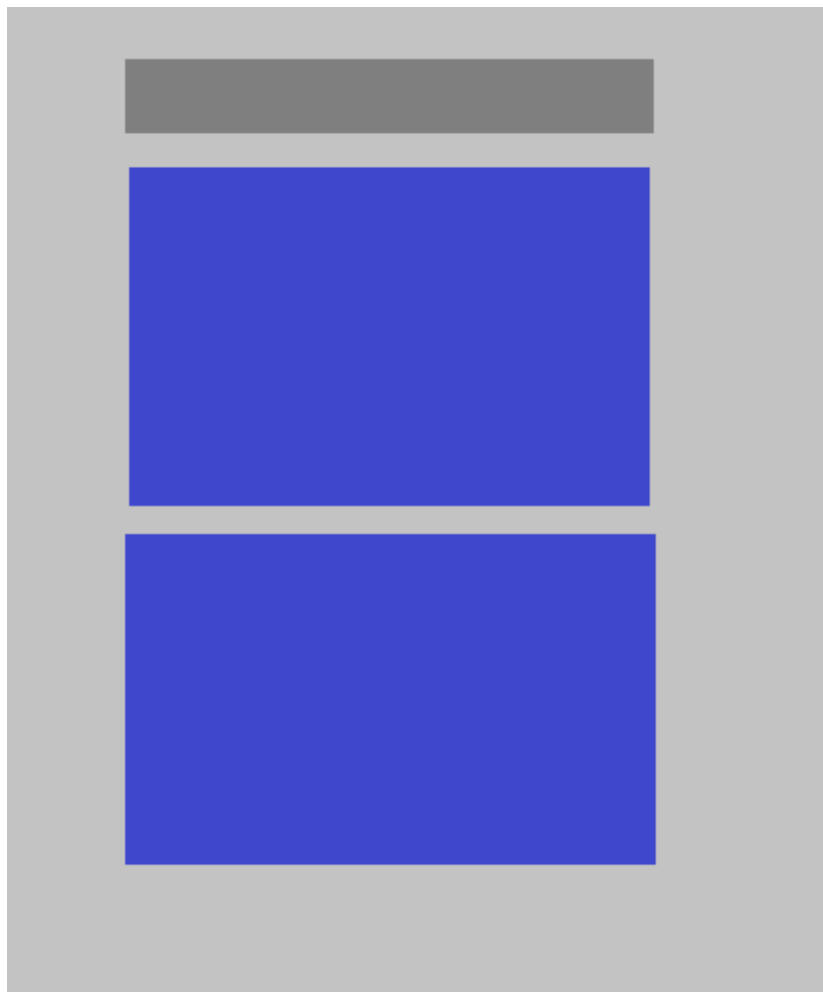


➤ **Cabecera:**

En la cabecera aparecen los elementos corporativos como el logo de la empresa y accesos rápidos.

➤ **Contenido:**

Este espacio está destinado a ser dos bloques centrales que contienen todo el grueso del contenido.



1.4 ESTILO VISUAL

1.4.1 TIPOGRAFÍA

Se ha utilizado Arial debido a la legibilidad en web.

Arial

abcdefghijklmnopqrstuvwxyzáéíóü
ABCDEFGHIJKLMNOPQRSTUVWXYZÁÉÍÓÜ 1234567890 \$%&(",;:'"!?)

Se ha utilizado Montserrat para el texto del slider

Montserrat

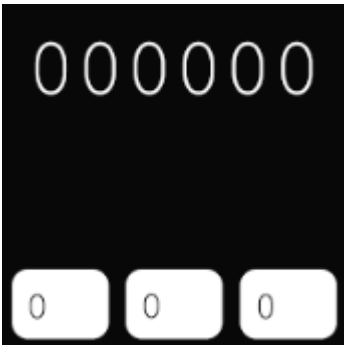
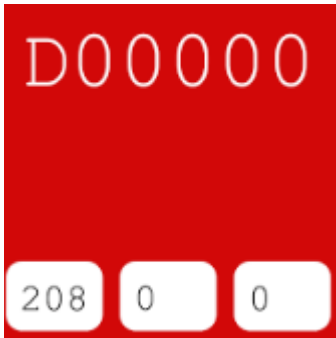
Abdcefg hijklmnñopqrstuvwxyzáéíóü
ABCDEFGHIJKLMNOPQRSTUVWXYZÁÉÍÓÜ 1234567890 \$%&(",;:'"!?)

1.4.2 COLORES

La página utiliza como predominio el gris y para contrastar utilizamos el blanco y el naranja.

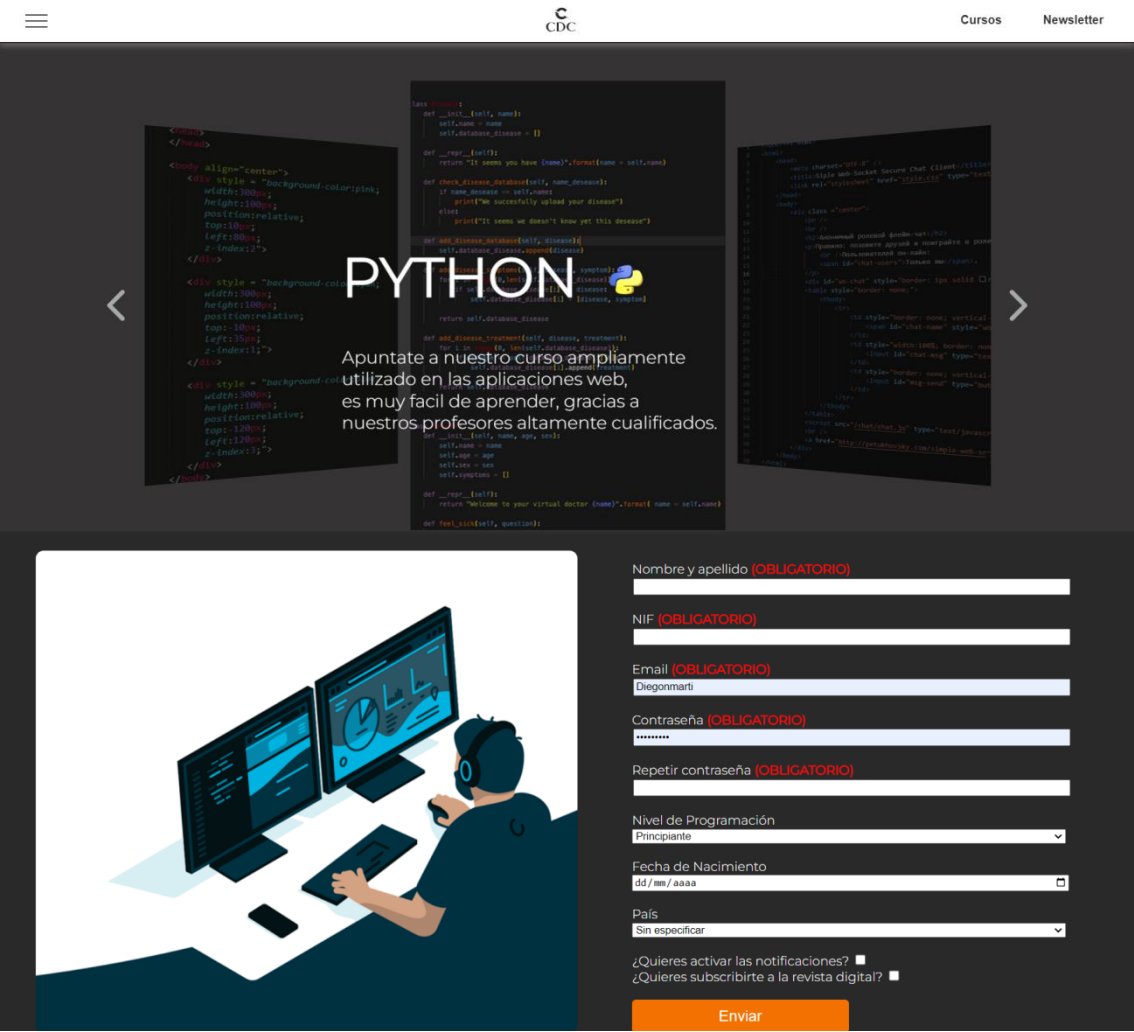


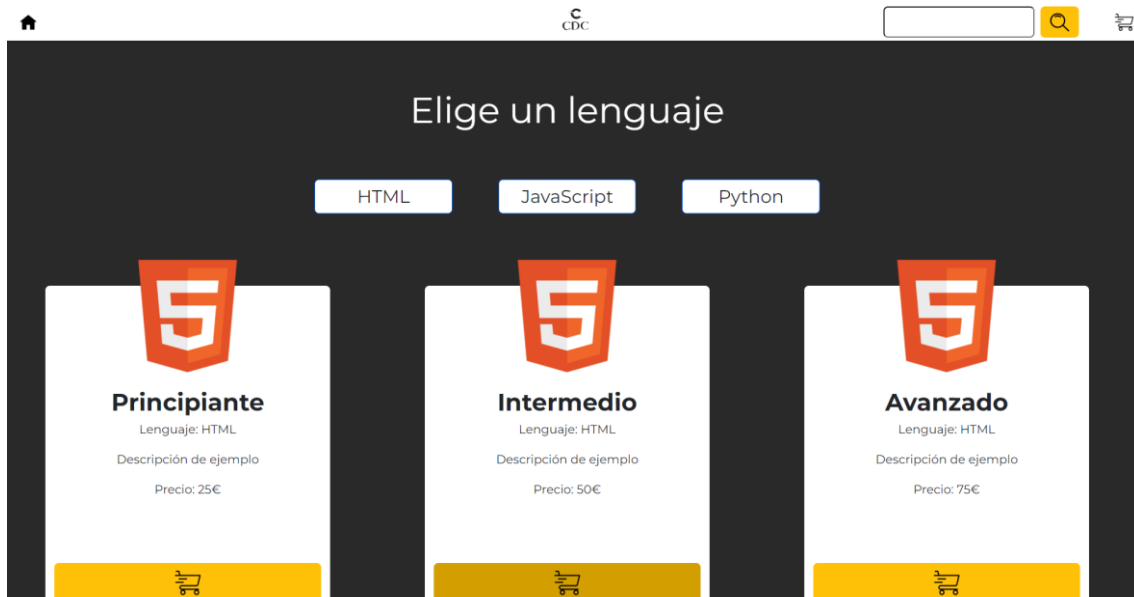
Colores secundarios:



1.5 RESULTADO LANDING PAGE

Este es el resultado final de nuestra Landing Page:





Buscador:

```
document.addEventListener("DOMContentLoaded", function() {
```

Método que permitirá frenar la ejecución del código hasta que toda la página esté cargada.

```
var valorBuscadoInput = document.getElementById("valorBuscado");
```

Variable que guarda el valor buscado en el Input

```
var botonBuscador = document.getElementById("botonBuscador");
```

Variable que guarda el botón del buscador.

```
valorBuscadoInput.addEventListener("input", function()
```

Se añade un escuchador de eventos al input.

```
var valorBuscado = valorBuscadoInput.value.toLowerCase();
```

Variable que guarda el valor buscado en el input en minúsculas.

```
if (/^(principiante|intermedio|avanzado)$/i.test(valorBuscado)) {
```

Si el valor coincide con uno de los niveles

```
valorBuscadoInput.style.border = "none";  
botonBuscador.disabled = false;
```

Se elimina el borde del input que indica un fallo y se habilita el botón para buscar.

```
} else {  
    valorBuscadoInput.style.border = "3px solid red";  
    botonBuscador.disabled = true;  
}
```

Si no coincide la búsqueda con los niveles se marca el fallo con un borde rojo en el input y se inhabilita el botón

```
botonBuscador.addEventListener("mouseenter", function() {
```

Se añade un escuchador al botón del buscador.

```
var valorBuscado = valorBuscadoInput.value.toLowerCase();
```

Variable que guarda el valor buscado en el input en minúsculas.

```
if (/^(principiante|intermedio|avanzado)$/.test(valorBuscado)) {  
    valorBuscadoInput.style.border = "none";  
    botonBuscador.disabled = false;  
  
} else {  
    valorBuscadoInput.style.border = "3px solid red";  
    botonBuscador.disabled = true;  
}
```

Mismo if que previamente.

2 GITHUB

<https://github.com/Diegonmarti/Proyecto-Retos>

The screenshot shows the GitHub repository page for 'Proyecto-Retos' by 'Diegonmarti'. The repository is public and has 2 branches and 0 tags. A warning message states 'Your main branch isn't protected'. The repository contains a table of files and folders with their commit history. The README.md file is visible, showing the project title 'PROYECTO RETOS' and a description of the project. The right sidebar contains information about the repository, including the README, activity, stars, and releases.

File/Folder	Commit Message	Time Ago
Cliente/DI1	Añadiendo Formulario a Cliente DI1 y LandingPage Version1 a Interface...	2 weeks ago
Despliegue App Web	Actualizando carpetas y DI1	last month
Documentacion	Añadiendo documentacion	20 hours ago
Empresa	Añadir carpeta moviles	5 days ago
Interfaces	version V9 de la landing interfaces DI6	19 hours ago
Moviles	Descripción de tus cambios	5 days ago
Servidor	Actualizando carpetas y DI1	last month
Videojuegos	Añadiendo documentacion	20 hours ago
README.md	Update README.md	37 minutes ago
gh-pages	Descripción de tus cambios	5 days ago
logo.png	Subiendo Sketch Landing DI4	last month

PROYECTO RETOS

Pincha para ver la landing page de nuestra página web.

1. Videojuegos

En este proyecto hay un video con los escenarios, además del terreno en sí, una oficina o zona donde se llevan a cabo tareas laborales.

- Un Escenario de un domicilio.
- Un Escenario como zona de oficina.
- Dos Escenarios con zonas naturales o de ocio.

Cada uno de los escenarios tiene dos texturas e incorporar árboles y alguna zona de agua. Además, se ha añadido una zona de viento (Wind Zone) y Assets de la store.

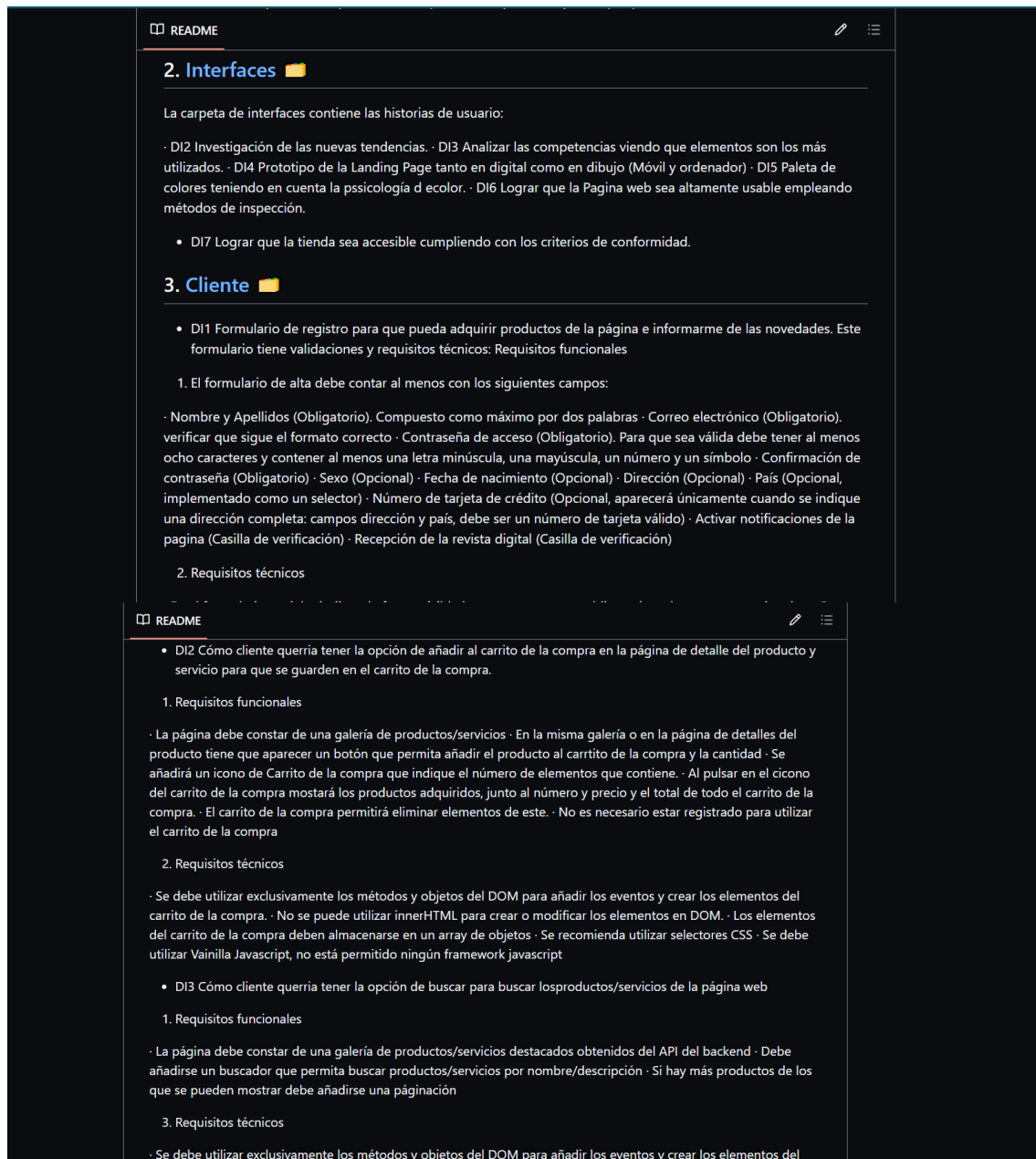
Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

Contributors
Diegonmarti Diego González
Code-Star2 CodeStar

Deployments
github-pages 5 days ago
+ 16 deployments

Languages
JavaScript 36.2% CSS 35.9%
HTML 19.8% Java 8.1%



Se ha modificado el GitHub de la siguiente manera:

- 1) Rama main: esta rama contiene una carpeta por cada asignatura y en cada asignatura las tareas correspondientes.
- 2) Rama landing-page: los archivos de la página web.

La rama main también contiene un Readme explicando con detalle cada carpeta del repositorio y cada título tiene un enlace que lleva a cada asignatura con sus respectivos ficheros.

Por último, se ha hecho uso de GitHub Pages para que desde el readme se pueda acceder directamente a la landing page. (En el siguiente video se ve como se abren)



video.mp4

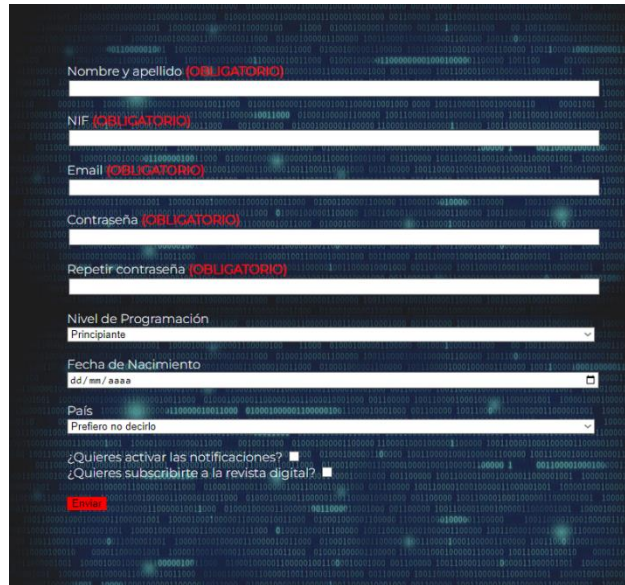


3 CLIENTE

3.1 Formulario HTML

Hemos añadido un formulario con varios inputs:

Nombre y apellido, NIF, Email, Contraseña, Repetir contraseña, Nivel de programación, fecha de nacimiento, país y dos checkbox.



The image shows a registration form with the following fields and elements:

- Nombre y apellido (OBLIGATORIO):** A text input field.
- NIF (OBLIGATORIO):** A text input field.
- Email (OBLIGATORIO):** A text input field.
- Contraseña (OBLIGATORIO):** A text input field.
- Repetir contraseña (OBLIGATORIO):** A text input field.
- Nivel de Programación:** A dropdown menu with "Principiante" selected.
- Fecha de Nacimiento:** A date picker showing "dd/mm/aaaa".
- País:** A dropdown menu with "Prefiero no decirlo" selected.
- ¿Quieres activar las notificaciones?:** A checkbox.
- ¿Quieres subscribirte a la revista digital?:** A checkbox.
- Enviar:** A red button at the bottom left.

Los campos Nombre y Apellido, NIF, Email, Contraseña y Repetir Contraseña son de tipo texto y deben ser completados obligatoriamente.

En caso de estar vacíos, se mostrará un mensaje de error. Los campos Nivel de Programación y País son de tipo select. El campo Fecha de Nacimiento es de tipo fecha.

A continuación, se encuentran dos casillas de verificación y el botón de envío (Submit).

3.2 Validaciones JS

```
function validarNombre(input) {
    var valor = input.value.trim();
    var nombreObligatorio = document.getElementById("nombreObligatorio");
    var nombreApellido = document.getElementById("nombreApellido");
    var nombreEtras = document.getElementById("nombreEtras");
    var nombreBorde = document.getElementById("nombreBorde");
    var iconNombre = document.getElementById("iconNombre");

    // Establecer clases de borde
    nombreBorde.classList.remove("borde-rojo");
    nombreBorde.classList.remove("borde-verde");

    // Validar y aplicar estilos
    if (valor == "") {
        nombreObligatorio.style.display = "block";
        nombreApellido.style.display = "none";
        nombreEtras.style.display = "none";
        nombreBorde.classList.add("borde-rojo");
        iconNombre.className = "icono icono-incorrecto";
    } else if (/^[a-zA-Z]{2,}$/.test(valor)) {
        nombreObligatorio.style.display = "none";
        nombreApellido.style.display = "none";
        nombreEtras.style.display = "none";
        nombreBorde.classList.add("borde-rojo");
        iconNombre.className = "icono icono-incorrecto";
    } else if (/^[a-zA-Z]{2,}$/.test(valor)) {
        nombreObligatorio.style.display = "none";
        nombreApellido.style.display = "none";
        nombreEtras.style.display = "none";
        nombreBorde.classList.add("borde-rojo");
        iconNombre.className = "icono icono-incorrecto";
    } else {
        nombreObligatorio.style.display = "none";
        nombreApellido.style.display = "none";
        nombreEtras.style.display = "none";
        nombreBorde.classList.add("borde-verde");
        iconNombre.className = "icono icono-correcto";
    }
}
```

En el campo “Nombre y apellido” hemos creado una función en la que si se pone menos de dos palabras o más de dos palabras dará fallo, también fallará si esas palabras tienen números o símbolos para que no de error se necesita poner dos palabras y solo se pueden utilizar letras mayúsculas y minúsculas.

Para validar el campo “NIF” hemos utilizado una función en la que se tienen que añadir 8 números seguido de una letra, para validar la letra se ha hecho un cálculo en que se basa en obtener el resto de la división entera del DNI y el numero 23. Con el resto de la división recorre el array y escoge la letra correcta.

```
function validarNIF() {
    var nifInput = document.getElementById("nif");
    var nifBorde = document.getElementById("nifBorde");
    var iconNIF = document.getElementById("iconNIF");

    var nif = nifInput.value;
    var expresionRegularNIF = /^[0-9]{8}[a-zA-Z]$/;
    var relleno = nif.slice(0, 8);
    var correcto = expresionRegularNIF.test(nif);

    if (correcto) {
        var numero = parseInt(nif.substring(0, 8));
        var letra = nif.charAt(8).toUpperCase();
        var letras = "TRAGXMBNVCZQJWKL";
        correcto = letras.charAt(numero % 23) === letra;
    }

    document.getElementById("nifObligatorio").style.display = relleno ? "none" : "block";
    document.getElementById("nifBorde").style.display = correcto ? "none" : "block";

    // Establecer clases de borde
    nifBorde.classList.remove("borde-rojo");
    nifBorde.classList.remove("borde-verde");

    // Aplicar estilos
    if (relleno && correcto) {
        nifBorde.classList.add("borde-verde");
        iconNIF.className = "icono icono-correcto";
    } else {
        nifBorde.classList.add("borde-rojo");
        iconNIF.className = "icono icono-incorrecto";
    }

    return relleno && correcto;
}
```

```
function validarEmail() {
    var email = false;

    var emailInput = document.getElementById("email");
    var emailBorde = document.getElementById("emailBorde");
    var iconEmail = document.getElementById("iconEmail");

    emailInput.addEventListener("input", function () {
        var email = emailInput.value;
        var expresionRegularEmail = /^[a-zA-Z0-9]{1,50}@[a-zA-Z0-9]{1,50}\.([a-zA-Z]{2,})$/;
        var relleno = email.slice(0, 50);
        var correcto = expresionRegularEmail.test(email);

        // Establecer clases de borde
        emailBorde.classList.remove("borde-rojo");
        emailBorde.classList.remove("borde-verde");

        // Aplicar estilos
        if (relleno && correcto) {
            emailBorde.classList.add("borde-verde");
            iconEmail.className = "icono icono-correcto";
        } else {
            emailBorde.classList.add("borde-rojo");
            iconEmail.className = "icono icono-incorrecto";
        }

        document.getElementById("emailObligatorio").style.color = relleno ? "green" : "red";
        document.getElementById("emailBorde").style.color = correcto ? "green" : "red";

        email = relleno && correcto;

        if (email) {
            document.getElementById("emailObligatorio").style.display = "none";
            document.getElementById("emailBorde").style.display = "none";
        }
    });

    // Muestra mensaje de error al cargar la página
    document.getElementById("emailObligatorio").style.display = "block";
    document.getElementById("emailBorde").style.display = "block";

    return email;
}
```

Para que el campo “Email” sea válido hemos hecho una función que establece como correcto si contiene un @ un punto y no tiene espacios.

En el campo “Contraseña” creamos una función en la que la contraseña tiene que contener al menos 8 o más caracteres, también requerirá de una minúscula, una mayúscula, un número y un símbolo.

```
function validarContraseña() {
    var password = false;

    var contraseñaInput = document.getElementById("contraseña");
    var contraseñaBorde = document.getElementById("contraseñaBorde");
    var iconContraseña = document.getElementById("iconContraseña");
    var contraseñaObligatorio = document.getElementById("contraseñaObligatorio");

    contraseñaInput.addEventListener("input", function () {
        var contraseña = contraseñaInput.value;
        var longitud = contraseña.length >= 8;
        var minuscula = contraseña.toLowerCase().indexOf("a") >= 0;
        var mayuscula = contraseña.toUpperCase().indexOf("A") >= 0;
        var numero = /[0-9]/.test(contraseña);
        var simbolo = /[!@#$%^&*()_+{}|;:,.<?~`]/.test(contraseña);
        var relleno = contraseña.slice(0, 8);

        // Establecer clases de borde
        contraseñaBorde.classList.remove("borde-rojo");
        contraseñaBorde.classList.remove("borde-verde");

        // Aplicar estilos
        if (relleno && longitud && minuscula && mayuscula && numero && simbolo) {
            contraseñaBorde.classList.add("borde-verde");
            iconContraseña.textContent = "✔"; // Checkmark
            iconContraseña.style.color = "green"; // Adjust color as needed
        } else {
            contraseñaBorde.classList.add("borde-rojo");
            iconContraseña.textContent = "✖"; // Cross
            iconContraseña.style.color = "red"; // Adjust color as needed
        }

        return password;
    });
}
```

```
function validarRepetirContraseña() {
    var esValido = false;

    var contraseña = document.getElementById('contraseña');
    var repetirContraseñaInput = document.getElementById('repetirContraseña');
    var contraseñaRepetirObligatorio = document.getElementById('contraseñaRepetirObligatorio');
    var contraseñaRepetirCoincide = document.getElementById('contraseñaRepetirCoincide');
    var contraseñaRepetirBorde = document.getElementById('contraseñaRepetirBorde');
    var iconoContraseñaRepetir = document.getElementById('iconoContraseñaRepetir');

    repetirContraseñaInput.addEventListener('input', function () {
        var repetirContraseña = repetirContraseñaInput.value;
        var relleno = repetirContraseña !== '';
        var coinciden = contraseña.value == repetirContraseña;

        // Lógica de validación y actualización de UI
    });
}
```

En el campo “repetir contraseña” se ha hecho una función en la que si se pone la contraseña del campo anterior se pondrá como válido.

3.3 Comprobación HTML

En cada campo se puede observar si el usuario tiene bien puestos los datos:

Campo nombre y apellido:

Nombre y apellido (OBLIGATORIO) X

El campo debe estar relleno

Nombre y apellido (OBLIGATORIO) X

El nombre solo puede contener caracteres

Nombre y apellido (OBLIGATORIO) X

Debes introducir nombre y primer apellido únicamente

Nombre y apellido (OBLIGATORIO) ✓

John Doe

Campo NIF:

NIF (OBLIGATORIO)

12345678X

El formato del NIF es incorrecto

NIF (OBLIGATORIO)

12345678Z

NIF (OBLIGATORIO)

El campo debe estar relleno

Campo Email:

Email (OBLIGATORIO)

El campo debe estar relleno

Email (OBLIGATORIO)

johndoegmailcom

El campo debe estar relleno

El formato del email es incorrecto

Email (OBLIGATORIO)

johndoe@gmail.com

Campo contraseña:

Contraseña (OBLIGATORIO)

El campo debe estar relleno

La contraseña debe contener al menos 8 caracteres

La contraseña debe contener al menos una minúscula

La contraseña debe contener al menos una mayúscula

La contraseña debe contener al menos un número

La contraseña debe contener al menos un símbolo

Contraseña (OBLIGATORIO)

El campo debe estar relleno

La contraseña debe contener al menos 8 caracteres

La contraseña debe contener al menos una minúscula

La contraseña debe contener al menos una mayúscula

La contraseña debe contener al menos un número

La contraseña debe contener al menos un símbolo

Contraseña (OBLIGATORIO)

El campo debe estar relleno

La contraseña debe contener al menos 8 caracteres

La contraseña debe contener al menos una minúscula

La contraseña debe contener al menos una mayúscula

La contraseña debe contener al menos un número

La contraseña debe contener al menos un símbolo

Campo repetir contraseña:

Repetir contraseña (OBLIGATORIO)

El campo debe estar relleno

La contraseña debe coincidir

Repetir contraseña (OBLIGATORIO)

El campo debe estar relleno

La contraseña debe coincidir

Repetir contraseña (OBLIGATORIO)

..... ✓

3.4 Carrito

Carrito

Nombre del curso: **JavaScript**

Nivel del curso: **Intermedio**

Precio del curso: **50€**

Eliminar

Nombre del curso: **HTML**

Nivel del curso: **Intermedio**

Precio del curso: **50€**

Eliminar

Total: **05050€**

```
var total = 0;
```

Variable de nombre total para el posterior cálculo del precio total a pagar de los productos en el carro.

```
var carrito = JSON.parse(localStorage.getItem('carrito'))
```

Variable en la que se guardan los datos del carrito recogidos del almacenamiento local

```
document.addEventListener('DOMContentLoaded', function () {
```

Método que permitirá frenar la ejecución del código hasta que toda la página esté cargada.

```
var botones = document.querySelectorAll('.curso-btn');
```

Variable en la que se guardarán todos los botones con la clase curso-btn.

```
botones.forEach(function (boton) {
  boton.addEventListener('click', function () {
```

Bucle for each que asignará una función a cada botón del array botones.


```
var datosCurso = JSON.parse(this.dataset.curso)
```

Variable que guarda los datos del curso asociados al atributo data-curso de los botones.

```
var nombreCurso = datosCurso.Lenguaje;  
var nivelCurso = datosCurso.Nivel; //  
var precioCurso = datosCurso.precio; /
```

Variables que guardan los valores correspondientes a los cursos de la variable datosCurso.

```
total += precioCurso;
```

Se le suma el precio del curso al total.

```
carrito.push(datosCurso);
```

Se añaden los datos del curso a la variable carrito.

```
var modalBody = document.querySelector('#carritoModal .modal-body');
```

Variable que guarda el modal del carrito.

```
var cursoElemento = document.createElement('div');
```

Variable que crea un div.

```
cursoElemento.innerHTML =  
'<p>Nombre del curso: <b>' + nombreCurso + "</b>" +  
'<br>Nivel del curso: <b>' + nivelCurso + "</b>" +  
'<br>Precio del curso: <b>' + precioCurso + "€</b>" +  
'<br><button class="eliminar-curso bg-warning rounded p-2 col-3 text-center d-flex">Eliminar</button><br><hr>';
```

Se le añade el texto al div anterior.

```
modalBody.appendChild(cursoElemento);
```

Se le añade como hijo el div al modal.

```
cursoElemento.querySelector('.eliminar-curso').addEventListener('click', function () {
```

Se añade un controlador de eventos al botón de eliminar.

```
total -= precioCurso;
```

Se resta el precio del total;

```
modalBody.removeChild(cursoElemento);
```

Se elimina el div correspondiente al curso añadido.

```
var index = carrito.indexOf(datosCurso);
```

Variable que guarda el índice del curso en el array del carrito.

```
if (index > -1) {  
    carrito.splice(index, 1);  
}
```

Si el curso está en el carrito, lo elimina.

```
localStorage.setItem('carrito', JSON.stringify(carrito));
```

Se guarda el contenido de la variable carrito en el almacenamiento interno, con el nombre también de carrito.

```
mostrarTotal();
```

Se llama a la función mostrar total.

```
carrito.forEach(function (datosCurso) {
```

Se realiza el mismo código de antes pero esta vez en lugar de para cada botón es para cada elemento guardado en el almacenamiento interno.

```
function mostrarTotal() {
```

Función para mostrar el total tras añadir o eliminar productos del carrito.

```
var modalFooter = document.querySelector('#carritoModal .modal-footer');  
modalFooter.innerHTML = 'Total: <b>' + total + '€</b>';
```

Variable que guarda el footer del modal y lo modifica concatenando el texto con la variable total.

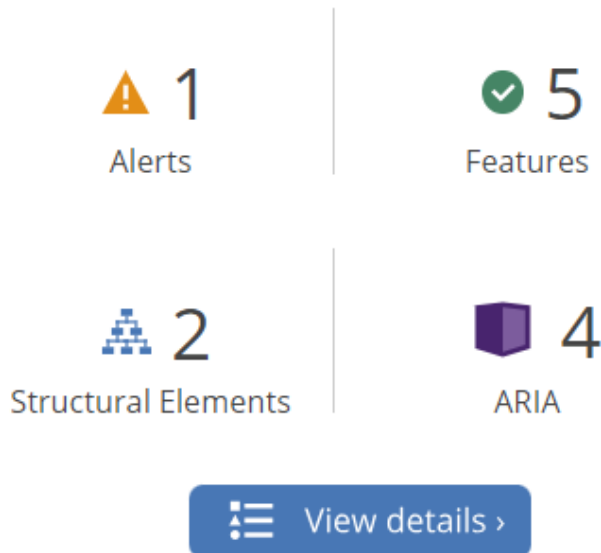
3.5 Interfaces

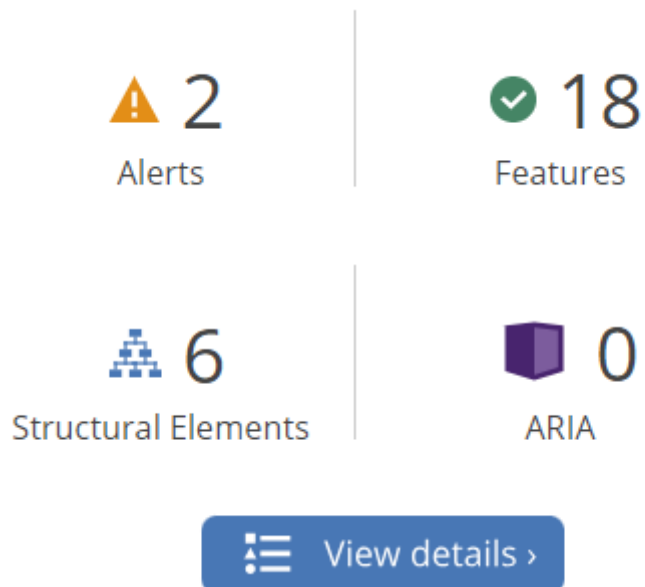
3.5.1 DI6

Lograr que la Página web sea altamente usable empleando métodos de inspección.

Hemos intentado que nuestra página sea usable siendo consistentes con los colores de la página, que se vea claro lo que queremos vender. Estéticamente, hemos intentado hacerlo atractivo y simple. Se intenta guiar a los clientes dejando claro con palabras e iconos que pueden hacer en la página web.

Hemos hecho uso de la extensión para Google: WAVE que nos dice los errores y alertas para que la estructura de nuestra página sea correcta. Además, te avisa de si la página que se ha desarrollado cumple con los criterios de evaluación WACG, en nuestro caso sí que cumplen con el nivel A, pero en el AA nos da problemas de usabilidad que debemos añadir. (Se prevee como unas mejores en un futuro).





3.5.2 DI7

Lograr que la tienda sea accesible cumpliendo con los criterios de conformidad.

En nuestro caso hemos intentado cumplir con los criterios de conformidad para llegar a un Nivel A.

Hemos priorizado que nuestra página se pueda utilizar para móviles, para que la gente con discapacidad que no pueda utilizar el ordenador pueda ver la página en móviles.

Se ha hecho que la página sea legible para los usuarios con discapacidad de la vista.

Se ha hecho un diseño claro y sencillo para que aparezca la información de manera fácil.