14 de mayo de 2023

Alberto Fernández Gutiérrez

UO265181

Universidad de Oviedo

SEW ENTREGA EXTRAORDINARIA

BITÁCORA

Contenido

[Consideraciones previas 2](#_Toc135137141)

[Concejo 2](#_Toc135137142)

[GitHub 2](#_Toc135137143)

[Archivos 2](#_Toc135137144)

[HTML 3](#_Toc135137145)

[CSS 4](#_Toc135137146)

[XML 4](#_Toc135137147)

[Computación en el cliente 10](#_Toc135137148)

[Computación en el servidor 13](#_Toc135137149)

[Pruebas 17](#_Toc135137150)

[XAMPP 17](#_Toc135137151)

[Nube (Azure) 17](#_Toc135137152)

[Validadores W3C 17](#_Toc135137153)

[Usabilidad 17](#_Toc135137154)

[Adaptabilidad 17](#_Toc135137155)

[Accesibilidad 17](#_Toc135137156)

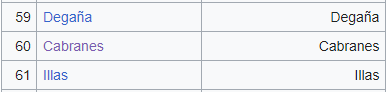
# Consideraciones previas

## Concejo

La elaboración del proyecto de extraordinaria comienza el 20 de abril. Ese día se obtiene el concejo sobre el que versará el trabajo:

UO265181 -> (265181 % 78) + 1 = 59 + 1 = 60

El concejo número 60 corresponde a Cabranes.



## GitHub

Se crea un repositorio en GitHub con el propósito de usar GitHub Pages y poder realizar un seguimiento del trabajo realizado, de esta forma, muchas de las entradas de la bitácora están acompañadas de un enlace al Commit realizado al repositorio.

<https://github.com/UO265181/sew-proyecto/>

El repositorio ahora mismo se encuentra público y el sitio web se despliega en el siguiente enlace:

<https://uo265181.github.io/sew-proyecto/entrega/proyecto/index.html>

**Pero:** es posible que partes de computación en el cliente no funcionen correctamente debido a que se usan claves de APIs que tienen usos límitados. Esto no ocurre con el proyecto entregado pues se han obtenido claves nuevas para la entrega (explicado con detalle en las entradas de Computación en el Cliente). Obviamente la parte de Computación en el Servidor no es funcional.

## Archivos

Se sigue la estructura de archivos marcada por el enunciado. Aclarar que dentro de la carpeta xml se guardan los archivos KML y SVG y que en la carpteta php los archivos de consultas a la base de datos.

# HTML

#### 20/04/2023 – 21:00

Se crea el archivo grastronomia.html en el que ya se incluyen una lista y una pequeña tabla con platos del concejo. También se añaden dos elementos de audio y uno de imagen a modo de prueba pues aún no están los archivos a los que se referencian. Es una primera versión de prueba para verificar el funcionamiento de GitHub Pages, que el marcado no es incorrecto y que se referencian bien a las hojas de estilo.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/8a52df42e5c9fefd70eea64fb96fc1eb3a7e1c1d>

##### Fuentes consultadas:

03-HTML5.pdf y tabla.html (campus virtual)

#### 23/04/2023 – 19:06

Avances en el html, se introducen varios metadatos (charset, autor, keywords…), se añade el nav (que después será replicado en el resto de documentos html), se completan los elementos de las tablas siguiendo ‘tabla-accesible.html’, una lista de definición con restaurantes de cabranes, texto sobre el festival del arroz con leche y una lista ordenada con los ganadores. También se agrega un footer que enlaza a mi dirección de correo de la universidad.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/5e2c7a277246fc3c243e9d3300f1b77627144a4c>

##### Fuentes consultadas:

03-HTML5.pdf, nav.html, tabla-accesible.html, footer.hmtl, address.html, anidar.html y metadatos.html (campus virtual)

<https://www.w3.org/WAI/WCAG21/Techniques/html/H48>

#### 30/04/2023 – 16:05

Se crea index.hmtl analógamente al gastronomía.html, pero solo contiene un breve texto sobre cabranas (además del nav y los metadatos). También se agregan la imágen de gastronomía.html así como el escudo de cabranes en svg para poder usarlo como icóno de las pestañas del sitio web.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/3db4119955ac12409ebbc8e2de5e7ce82d31e8d6>

##### Fuentes consultadas:

03-HTML5.pdf, imagenes.html (campus virtual)

<https://www.w3schools.com/html/html_favicon.asp>

#### 30/04/2023 – 19:46

Se finaliza gastronomia.html: Se le agrega el marcado semántico, se expande la tabla de platos, a la lista de definición de los restaurantes se le anida una lista no ordenada con sus mejores platos, se agrega un vídeo sobre el festival del arroz con leche y los archivos multimedia necesarios. Se fueron detectando errores usando el validador de W3C y corriguiendo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HTML | | | | |
|  | BODY | | | |
|  | HEADER | | |
| NAV | | |
| MAIN | | |
|  | SECTION | |
| SECTION | |
| SECTION | |
|  | ARTICLE |
| SECTION |
| SECTION |
| FOOTER | | |

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/55a3fd570545a435aad91d45c25636f074626b6e>

##### Fuentes consultadas:

03-HTML5.pdf, main.html, estructuraCompleta.html, section.html, article.html, audio.html y video.html (campus virtual)

<https://validator.w3.org/>

# CSS

#### 20/04/2023 – 21:00

Se crean dos archivos css, estilo.css y layout.css. De momento estos archivos solo son un borrador y su uso es meramente comprobar que estan enlazados al los documentos html de forma correcta.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/8a52df42e5c9fefd70eea64fb96fc1eb3a7e1c1d>

##### 

#### 30/04/2023 – 16:05

Ahora los archivos de estilo están personalizados para el sitio web y tienen una coherencia cromática y de tamaños. Para el layout se opta inicialmente por un grid.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/3db4119955ac12409ebbc8e2de5e7ce82d31e8d6>

##### Fuentes consultadas:

04-CSS-Conceptos.pdf (campus virtual)

<https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout>

<https://www.w3schools.com/css/css_grid.asp>

#### 30/04/2023 – 16:05

Al no acabar de conseguir encajar el grid layout con el concepto visual del sitio web se opta siemplemente por un display en block de los distintos elementos. Se separan las reglas de posicionamiento del resto y se detalla la especifidad de cada una de ellas.



##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/55a3fd570545a435aad91d45c25636f074626b6e>

##### Fuentes consultadas:

04-CSS-Conceptos.pdf (campus virtual)

<https://developer.mozilla.org/es/docs/Web/CSS/:hover>

# XML

#### 01/05/2023 – 14:54

Se crea el arhivo rutas.xml siguiendo la estructura del enunciado. El archivo aún esta falto de información pero contiente la estrucura necesaria para ir realizando los validadores DTD y Schema. Se crea también rutas.dtd con la mayor parte de reglas escritas.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/531846283fa270d59cdff950cce88c376b6c2232>

##### Fuentes consultadas:

01-Introducción-XML.pdf, 02-DTDs.pdf (campus virtual)

#### 01/05/2023 – 15:55

Se crea el arhivo rutas.xsd para validar la correcta estructura y contenido de rutas.xml. El archivo ya contiene la gran parte de reglas necesarias.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/fec73b7d6786f5da6d403d892a23cf9241ba5583>

##### Fuentes consultadas:

01-Introducción-XML.pdf, 03-Schemas.pdf, 04-Ejemplos.rar (campus virtual)

#### 01/05/2023 – 18:55

El primer aproximamiento para poder mostrar las rutas del fichero xml en el sitio web es crear el archivo rutas.html, el cúal conoce a rutas.js y llama a un método del mismo (rutas.verXML()) al pulsar un botón. De esta forma las rutas no se mostrarán (no se realizará la traducción a html) hasta que se pulse el botón. Esta no es la propuesta final, pero facilita la detección de errores y la depuración del código.

El archivo rutas.html contiene lo común al resto de html y el botón para cargar las rutas.

El archivo rutas.js se construye inicialmente siguiendo la estructura del archivo 90-jQuery-AJAX-XML-Oviedo.js del campus virtual. Por el momento solo se muestra el xml como una simple cadena de caracteres.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/ae728d9750f404d42351e9dc6f6646dcd8e943d5>

##### Fuentes consultadas:

06-Procesamiento-XML.pdf, 04-Ejercicios-JS.zip (campus virtual)

#### 01/05/2023 – 19:00

Se corrigue un error, en la etiqueta script de rutas.html, la url usada era http y no https.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/f911c3d76aa39a5a172e255d32367a68fb5c2dcc>

#### 02/05/2023 – 16:14

Ahora se muestra el nombre de una de las rutas al pulsar el botón.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/ffc21e431ef0ec6ee9872b69ea8bf881f3a7a84c>

##### Fuentes consultadas:

3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

#### 02/05/2023 – 16:40

Ahora se muestra el nombre de las rutas, y para cada una de ellas una lista con algunos de sus atributos.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/38f2de494035d4f3e19b571d794287260a88d10b>

##### <https://github.com/UO265181/sew-proyecto/commit/12d853650191ff8bbc9ef8fb6a72abd16da2dc82>

##### <https://github.com/UO265181/sew-proyecto/commit/f55c7d8462f6d47df541797d66b217d08fb093fa>

##### Fuentes consultadas:

3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

#### 02/05/2023 – 17:05

Se corrigue un pequeño error (variables con nombre incorrecto) y se crea un bucle que recorre los hitos de cada ruta traduciendo la información de estos.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/109c115803556c5fe3b34e1c197c91f2a406b6aa>

##### <https://github.com/UO265181/sew-proyecto/commit/7c0602a4a57ffea670282638f29eb57b40fd55bd>

##### <https://github.com/UO265181/sew-proyecto/commit/418f54cc0aa1f26de98398fa66a958a93bc56790>

##### Fuentes consultadas:

3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

#### 03/05/2023 – 20:23

Se rellena la infomación de las dos primeras rutas del xml (descripciones, imágenes…).

Se añade la comprobación en rutas.dtd y ruta.xsd de que las imágenes del xml han de contener un atributo alt y otro figCaption para poder usar esa información en la traduccion y evitar problemas de accesibilidad.

En rutas.js se añade el código necesario para mostrar la mayoría de la información de las rutas y sus hitos (faltan la altímetria y la planimetría). También se borra código muerto que no se estaba utilizando.

En rutas.xsd se añaden las reglas necesarias para hacer posible que la fecha y la hora de comienzo de una ruta sea un elemento opcional.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/fce3840a1f7939145dae83f60dd8c019bf438e64>

##### Fuentes consultadas:

02-DTDs.pdf, 03-Schemas.pdf, 3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

<https://es.wikiloc.com/>

<https://www.turismoasturias.es/>

#### 04/05/2023 – 11:12

En rutas.js se añade el código restante para mostrar toda la información de las rutas, así como sus recursos multimedia.

En rutas.xsd se añade la restricción de que las etiquetas de video han de contener un atributo ‘type’.

En rutas.xml se mejora la información (puntuación, atributo ‘type’ en los videos…).

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/f16ab744ed2494584f8f5d95461f4597c81d9ff5>

##### Fuentes consultadas:

03-Schemas.pdf, 3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

#### 04/05/2023 – 19:12

Siguiendo la documentación de W3C se realiza un primer borrador del traductor a kml (de momento no funcional) y se elimina más código muerto.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/7d8cb4962e580c7a1f764687549412149e8dc560>

<https://github.com/UO265181/sew-proyecto/commit/531a1985b0be54d6952989d49272d668c24c2f61>

##### Fuentes consultadas:

<https://www.w3schools.com/xml/dom_document.asp>

<https://www.w3schools.com/xml/met_document_createelement.asp>

#### 05/05/2023 – 13:05

Se añade ‘ida\_y\_vuelta’ como atributo de una ruta. De esta forma, al generar el SVG de la altimetría se pueden representar mejor las rutas de este tipo. Se añaden al DTD y al Schema las reglas relativas.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/7292eefdf1fda7332ed87c16efef36a4834144be>

<https://github.com/UO265181/sew-proyecto/commit/7011daa0a69275eb88341d3b78583fc4647f80ff>

<https://github.com/UO265181/sew-proyecto/commit/419c1d5812a64db1b70c13e93e1bc43f13ae9607>

##### Fuentes consultadas:

02-DTDs.pdf, 03-Schemas.pdf (campus virtual)

#### 05/05/2023 – 18:42

Al no tener ciertas dudas sobre el archivo KML se crea un pregunta en el foro de dudas de extraordinaria del campus (<https://www.campusvirtual.uniovi.es/mod/forum/discuss.php?d=69019>) y se prosigue con el SVG.

En rutas.js se añade el código para generar el SVG de cada ruta:

* Según se recorren las rutas para traducirlas, se almacena en un array `puntosSVG’ las alturas.
* Si la ruta es de ida y vuelta, se añaden los puntos de la vuelta.
* Para evitar que los puntos no se encuentren fuera de los límites del archivo SVG y que este no sea muy pequeño o muy grande se opta por darle al archivo un tamaño fijo de 1000x1000px y escalar los puntos.
* Se añade un punto falso (se utiliza para el calculo del escalado pero no se muestra en el SVG) para que los puntos de la ruta no estén demasiado pegados a los bordes.
* Para cada ruta se crea el archivo SVG siguiendo la documentación de W3C y se van añadiendo los puntos. Se añade también una línea que los recorre.
* Finalmente, para cada ruta se crea un botón que descarga el SVG que se acaba de construir al cargar rutas.js y se muestra la dirección donde se almacena el SVG en el proyecto.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/38f37bee6b02edbebec7ee266855e376520f331e>

<https://github.com/UO265181/sew-proyecto/commit/2362d4ef2a16868f81efa7997c751f880ab4a22f>

##### Fuentes consultadas:

3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

<https://www.w3schools.com/xml/dom_document.asp>

<https://www.w3schools.com/xml/met_document_createelement.asp>

<https://developer.mozilla.org/es/docs/Web/SVG>

#### 13/05/2023 – 12:44

Para el KML de la planimetría de las rutas se sigue el mismo procedimiendo que con la altimetría. Mientras se traduce el xml se almacena en un array las coordenadas y luego estas se utilizan para crear los puntos de documento KML. También se añade el botón para descargar el archivo recién generado.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/dc7f657ec146e5549f77fcbf6974e8f876b8c38a>

<https://github.com/UO265181/sew-proyecto/commit/81a843298afd25046fc933f7b8687484f22b9877>

##### Fuentes consultadas:

3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

<https://www.w3schools.com/xml/dom_document.asp>

<https://www.w3schools.com/xml/met_document_createelement.asp>

<https://developers.google.com/kml/documentation/kml_tut?hl=es>

# Computación en el cliente

#### 05/05/2023 – 19:38

Se crea el archivo carrusel.js con el código necesario para mostrar las imágenes, se añade al index.html y se añaden cinco imágenes.

El archivo js contiene un array con la información de las imágenes y se encarga de crear una sección en la que hay un elemento img y figCaption. Haciendo uso de setInterval(), cada cierto tiempo, la imagen que contiene la sección varía al recorrerse el array. Se añaden también dos botones para ver la imagen anterior y la siguiente.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/da99fa398e02ec5564693c1497155d2139f49ff7>

##### Fuentes consultadas:

3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

<https://developer.mozilla.org/en-US/docs/Web/API/setInterval>

<https://www.w3schools.com/jsref/met_win_setinterval.asp>

#### 06/05/2023 – 13:36

Se crea el archivo tiempo.js con el código necesario para mostrar el tiempo actual en Cabranes y se añade al index.html.

Para el código se sigue la estructura de 86-jQuery-JSON-meteo.js del campus virtual. Es decir, se hace la llamada a la api, se recoge la información a mostrar y finalmente se genera el código html.

Se utiliza la api gratuita de OpenWeatherMap:

* Documentación: <https://openweathermap.org/current>
* Llamada: <https://api.openweathermap.org/data/2.5/weather?id=6359916&appid=f584ab7920bcc2b1c18704d726aea2a5&units=metric&lang=sp>
* Uso de clave: La clave permite 60 llamadas por minuto, por lo que no debería haber ningun problema con ella.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/d0328cd9b6a31f82344ce249b8abe0654e6a0d73>

##### Fuentes consultadas:

386-jQuery-JSON-meteo.js (campus virtual)

<https://openweathermap.org/current>

#### 07/05/2023 – 13:21

Se crea el archivo noticias.js con el código para mostrar tres noticias recientes sobre Cabranes y se añade al index.html. Es probable que se muestren noticias no muy específicas sobre el concejo en sí o que estas no sean demasiado recientes.

Para el código se sigue la misma estructura que en tiempo.js.

Se utiliza la api gratuita de Newscatcher:

* Documentación: <https://docs.newscatcherapi.com/>
* Llamada: <https://api.newscatcherapi.com/v2/search?q=Cabranes%20AND%20Asturias&lang=es&sort_by=date>
* La clave se especifica en la cabezera del GET.
* Uso de clave: Solo permite 50 llamadas en total, por lo que es posible que se acaben y no se mueste correctamente el contenido en GitHub Pages o en la máquina de Azure. En el código de la entrega está una nueva clave que solo ha sido usada una vez para comprobar su funcionamiento. Para la defensa (si se diese) se generará otra clave si es necesario. De todas formas, dejo aquí una clave sin ningún uso: XEEkQzeZAcTsTpAYHnyRFGy\_TkPSpQtUlWyHi\_ZJVAo

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/cd44aade4c94e508142e426d6cad006e00547ada>

<https://github.com/UO265181/sew-proyecto/commit/b0a5d85be171711f408ead7fadecd1512e4992ad>

##### Fuentes consultadas:

<https://docs.newscatcherapi.com/>

#### 07/05/2023 – 16:01

Se crea el archivo mapa.js con el código para mostrar un mapa de situación de Cabranes en Asturias.

Para el código se sigue la misma estructura que en el resto de archivos JavaScript de Computación en el Cliente.

Se utiliza la api gratuita de MapQuest:

* Documentación: <https://developer.mapquest.com/documentation/static-map-api/v5/map>
* Llamada: <https://www.mapquestapi.com/staticmap/v5/map?key=2LVGEKH0ZArUc3dX6Jj2nMLmJSqeLLaW&locations=Cabranes&size=1100,500@2x&type=hyb&scalebar=true&defaultMarker=circle-white&zoom=10>
* Uso de clave: Permite 15.000 consultas al mes.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/260631254c3ab0b71d5c867cc20b22e59501b45f>

##### Fuentes consultadas:

<https://developer.mapquest.com/documentation/static-map-api/v5/map>

<https://developer.mapquest.com/documentation/static-map-api/v5/getting-started/>

#### 08/05/2023 – 18:05

Se crea el archivo meteorologia.html y meteorologia.js con el código para mostrar la previsión del tiempo en Cabranes durante siete días (incluyendo el actual).

Se utiliza la api gratuita de WeatherBit:

* Documentación: <https://www.weatherbit.io/api/weather-forecast-16-day>
* Llamada: <https://api.weatherbit.io/v2.0/forecast/daily?key=91e4b6637f7f40e3a5b398092e6a4f66&lang=es&units=M&days=7&lat=43.407598&lon=-5.415428>
* Uso de clave: Permite 50 llamadas diarias.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/0cf0109bf7fdb5542a275ffb86b6fb1a192efd0f>

##### Fuentes consultadas:

<https://www.weatherbit.io/api>

<https://www.weatherbit.io/api/weather-forecast-16-day>

#### 08/05/2023 – 19:10

Se crea el archivo juego.html y juego.js.

Es el archivo JavaScript el que se encarga de escribir el html del formulario con las preguntas del juego y calcular las respuestas acertadas. En este archivo también se encuentran las preguntas que se muestran (así como sus respuestas). Todas las preguntas versan sobre la información de Cabranes que se muestra en el sitio web.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/3cce5aafc0ffa666559ad8976ac99d0b12ca3ebb>

##### Fuentes consultadas:

3-El-lenguaje-JavaScript.pdf, 04-Ejercicios-JS.zip (campus virtual)

#### 08/05/2023 – 19:11

Se revisa y corrigue el código de los distintos archivos de esta parte del proyecto para que cumplan el POO.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/08c7ed320fa472a3ce69e414de29b4d24c936b38>

##### <https://github.com/UO265181/sew-proyecto/commit/c4d9fcaeb21a418dd994261c15ff1823cd1384c4>

<https://github.com/UO265181/sew-proyecto/commit/45cf91c8b41b219ecf8017158220865a1c0dc559>

<https://github.com/UO265181/sew-proyecto/commit/a7f0ef4b073532469025f60ad103eeb2530c93cf>

<https://github.com/UO265181/sew-proyecto/commit/4729ef79f0b437d63ed5c75d21fd99ff752bbeef>

##### 

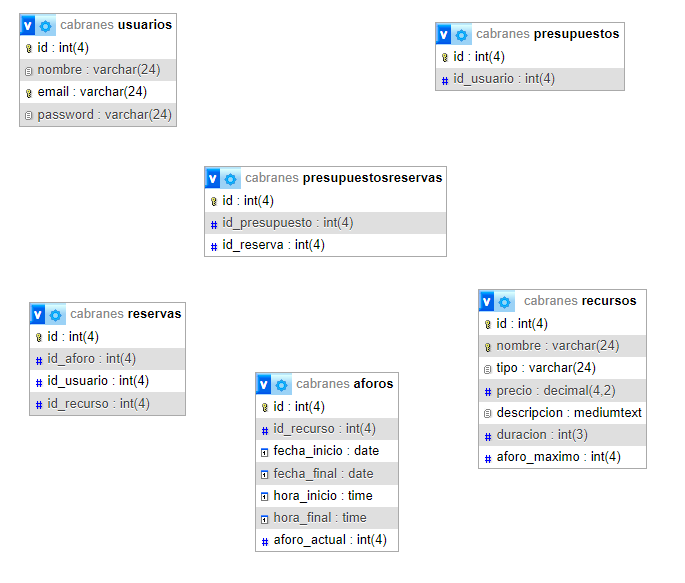
# Computación en el servidor

#### 09/05/2023 – 15:20

Se empieza por realizar un borrador de la base de datos requerida para la realización del proyecto.

##### 

En la base de datos real se utiliza una tabla adicional (presupuestosReservas) para resolver la relacion n-n entre presupuestos y reservas.



##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/b31b01697f8640a76c14f26ed773211e4b63e660>

#### 09/05/2023 – 19:21

Se crea el archivo reservas.html con la idea de tener ahí los distintos formularios (registro, reserva, indentificación…) y que estos llamen (al hacer submit) al código php que se encargue de procesarlos.

Se añade el formulario de registro y de inicio de sesión en reservas.html.

Se crea el archivo registrarse.php con el código para gestionar la respuesta al post del formulario correspondiente. Se validan los datos de entrada y se inserta al usuario en la base de datos para que pueda identificarse.

Se crea el archivo inicarSesion.php con el código para gestionar la respuesta al post del formulario correspondiente. Se validan los datos de entrada y se comprueba si existe un usuario con tales datos y si es así se identifica al usuario haciendo uso de $\_SESSION.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/8d8f8005e247afd07e7f2dd3fbf7b4e299f4b1aa>

##### Fuentes consultadas:

2-El-lenguaje-PHO.pdf, 3-PHP-MySQL.pdf, SEW-php.rar (campus virtual)

#### 10/05/2023 – 18:04

Para seguir la programación orientada a objetos, se crean dos nuevos archivos:

* conexionDB.php: se encarga de realizar las consultas, abrir y cerrar la conexión con la base de datos.
* validacion.php: se encarga de comprobar el formato de los datos que se utilizarán en las consultas.

Ahora al realizar el registro se comprueba que el email no exista ya en la base de datos.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/89658c994cc6e4dad801d0342cdec311179e3c2b>

##### Fuentes consultadas:

2-El-lenguaje-PHO.pdf, 3-PHP-MySQL.pdf, SEW-php.rar (campus virtual)

#### 11/05/2023 – 14:39

Se renombra el archivo iniciarSesion.php a identificarse.php y se termina el código para que el registro y la identificación sea funcional. Para ello:

* Se deja de usar reservas.html y en su lugar el navegador del sitio web redireccionará a reservas.php.
* El archiv reservas.php contiene lo común a los demás html del sitio web y en main código php que gestiona los post y define la vista de la página dependiendo de si el usuario se ha identificado o no. Los encargados de mostrar los formularios son las respectivas clases de los archivos identificarse.php y registrarse.php.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/233edda36847384fe56719c852c0ae6d819e2a3f>

##### Fuentes consultadas:

2-El-lenguaje-PHO.pdf, 3-PHP-MySQL.pdf, SEW-php.rar (campus virtual)

#### 11/05/2023 – 19:27

Se unifica identificarse.php y registrarse.php en el archivo usuarios.php, y en la clase Usuarios. Se crea el archivo recursos.php para la gestión de recursos (muestra una tabla con los recursos existentes en la base de datos).

Con esta mejora de diseño se tiene un archivo/clase por cada tabla.

##### Commit:

##### <https://github.com/UO265181/sew-proyecto/commit/7abbcfe8566d5d580e3fa8287decda0d3ec8f5c9>

##### Fuentes consultadas:

2-El-lenguaje-PHO.pdf, 3-PHP-MySQL.pdf, SEW-php.rar (campus virtual)

#### 12/05/2023 – 18:36

Se crea el código para la realización de reservas. En el archivo reservas.php se gestiona si el usuario está identificado, si es así llama a Recursos para que muestre/reciba el formulario. En el formulario se ha de especificar nombre, fecha y hora del recurso. Si el recurso tiene el aforo completo para esa fecha y hora la reserva no se realiza y el usuario es notificado.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/56ee4a10fba9f444a5f678bf1d487aeb536e8431>

<https://github.com/UO265181/sew-proyecto/commit/6ed0f2b75225aa97f74530a14df5af8ad86c7889>

<https://github.com/UO265181/sew-proyecto/commit/f026998244294f1b78d67486da421ec06965c32f>

<https://github.com/UO265181/sew-proyecto/commit/9c79c4e30c10e8ac77baad0f93dbfab6c6bc81d2>

##### Fuentes consultadas:

2-El-lenguaje-PHO.pdf, 3-PHP-MySQL.pdf, SEW-php.rar (campus virtual)

#### 12/05/2023 – 18:57

Se añade el código para mostrar en la página una tabla de todas las reservas realizadas por el usuario.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/7838eb5d1236b0924ef2166710ccef063b199875>

#### 13/05/2023 – 11:19

Se crea el archivo presupuestos.php para la gestión de los presupuestos. Se añade un botón a la página que se encarga de generar el presupuesto:

* Si el usuario no ha realizado reservas, no se genera el presupuesto y se le notifica
* Si el usuario ha realizado reservas, se genenera un archivo de texto (que se le descarga al usuario automáticamente) con la descripción del presupuesto (usuario, reservas, precio total) y se inserta en la base de datos.

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/7ba4fe64b471c4f0be03b028ee3f2886543ad250>

<https://github.com/UO265181/sew-proyecto/commit/02b500a9d64ec8ae8a5c9ba37cf2e07f9dd2033f>

#### 16/05/2023 – 12:40

Para garantizar la POO en la computación en el cliente se extrae el código de reservas.php a vistas.php, se hace un encapsulamientos de los procedimientos en distintas funciones y se hace un renombrado de los archivos/clases. De esta manera se tiene el siguiente diseño en capas:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| reservas.php |  | vista.php |  | usuariosObject.php |  |  |
|  | conexionDB.php |
| reservasObject.php |
| -> | -> |  | -> |  |
|  |  | recursosObject.php |  | validacion.php |
|  |
| presupuestosObject.php |  |

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/8b915eb5809483762977f41b77600b05ad999737>

#### 16/05/2023 – 12:46

Se crean dos archivos sql con las consultas para crear la base de datos (cabranes.sql) y las filas de los recursos turísitcos (recursos.sql).

##### Commit:

<https://github.com/UO265181/sew-proyecto/commit/588af7ac1470f8b9943575fe54525eadfbf16401>

##### <https://github.com/UO265181/sew-proyecto/commit/bfb1e0a5c04659558546fce0ce5a48665df88abb>

##### Fuentes consultadas:

3-PHP-MySQL.pdf

# Pruebas

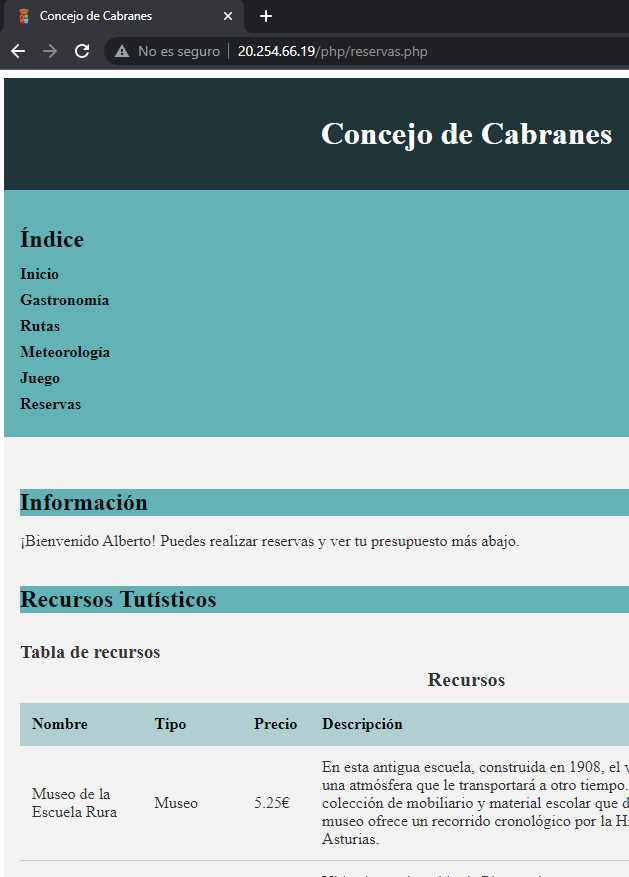
## XAMPP

Se instala XAMPP y se cambia el archivo de configuración de Apache para que apunte al directorio donde se encuentran los archivos de proyecto. Se deja el resto de configuración por defecto y se lanza Apache y MySQL.

Es desde este entorno donde se ha ido comprobando el funcionamiendo del sitio web durante el desarrollo. Para las consultas en MySQL se ha creado el usuario sewAdmin::sewAdmin+01, la base de datos cabranes (cabranes.sql) y los recursos turísticos de recursos.sql.



## Nube (Azure)

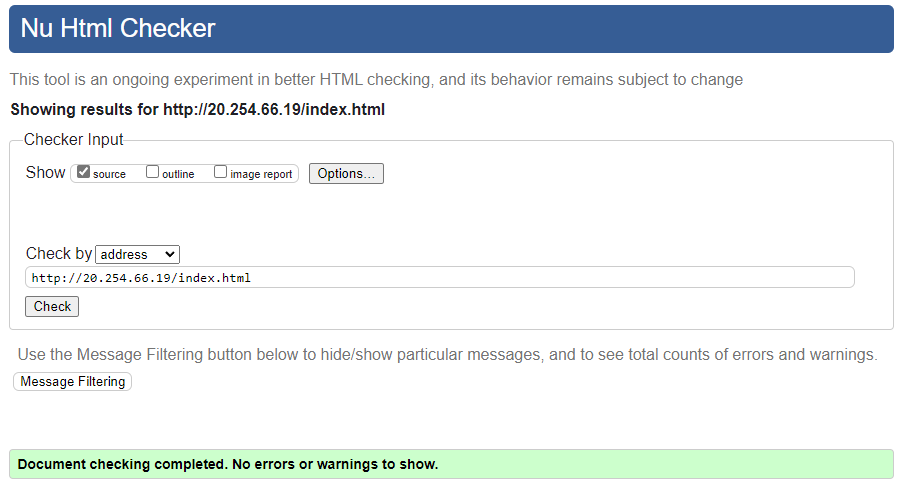
Siguiendo la guia de configuración de la máquina Ubuntu en la nube de Azure del campus (6-Azure-Servidor-Web.pdf), se instala el servidor apache, php y mysql. Luego usando sftp se vuelcan los archivos del proyecto. Después, mediante la línea de comando de mysql se crea el usuario para realizar las consultas de la aplicación (sewAdmin@localhost, sewAdmin+01), se crea la base de datos usando el archivo cabranes.sql y se insertan los recursos turísticos de recursos.sql. 

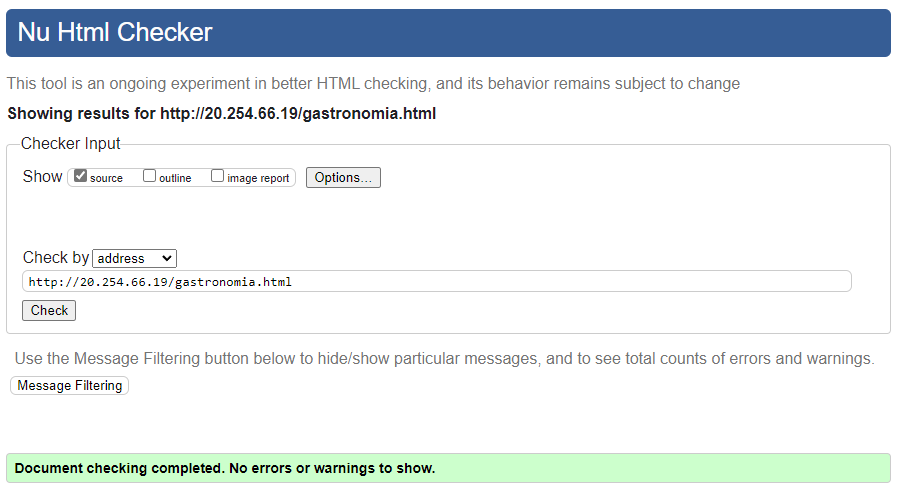
Se puede verificar su correcto funcionamiento en la siguiente dirección: <http://20.254.66.19/>

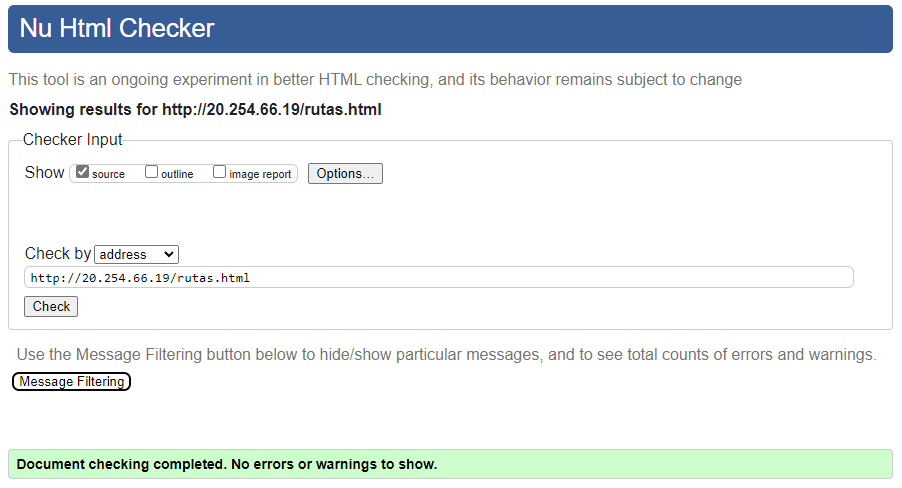
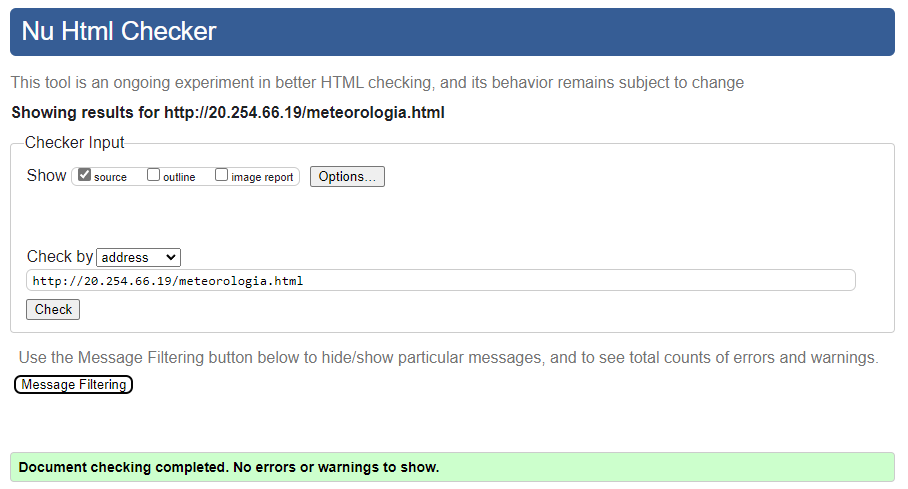
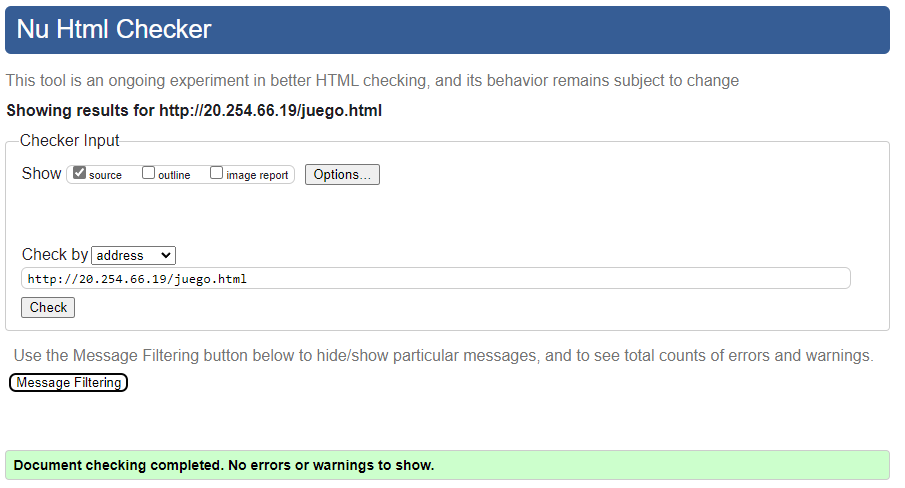
**Pero:** es posible que partes de computación en el cliente no funcionen correctamente debido a que se usan claves de APIs que tienen usos límitados. Esto no ocurre con el proyecto entregado pues se han obtenido claves nuevas para la entrega (explicado con detalle en las entradas de Computación en el Cliente). Llegado el momento de la defensa se renovarán las claves oportunas.

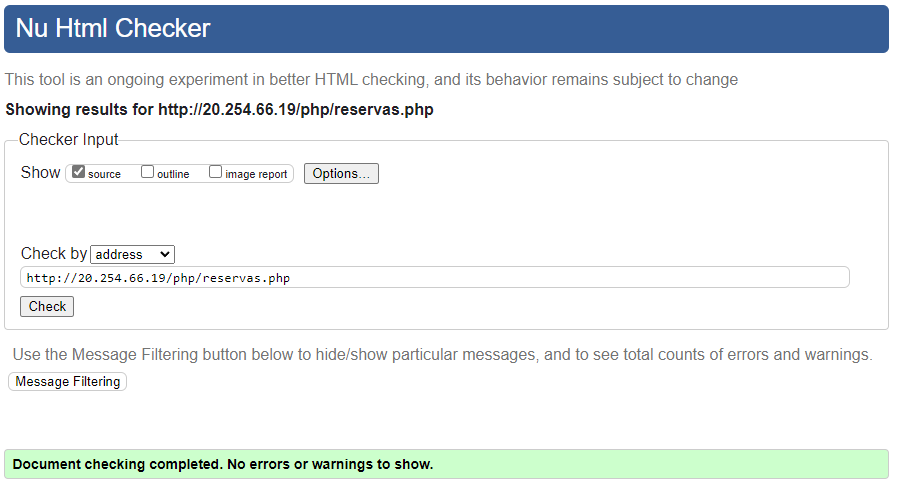
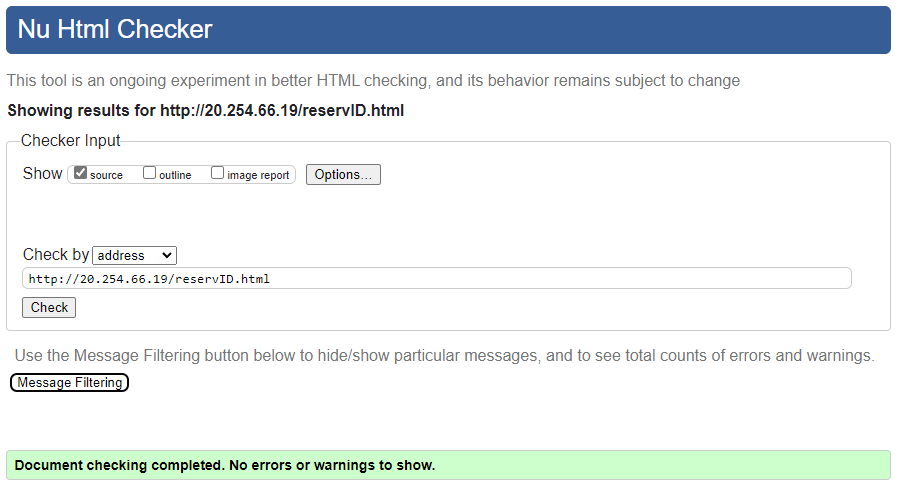
## Validadores W3C

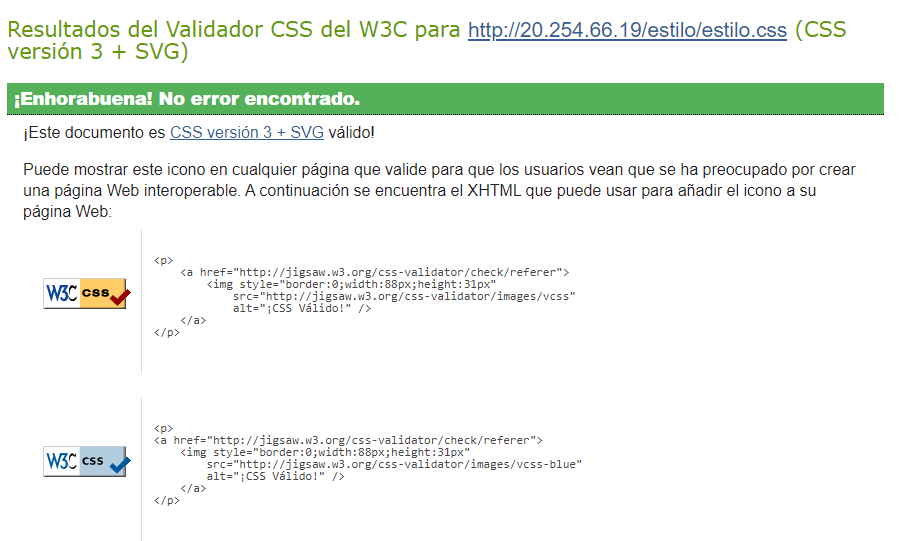
Todas las validaciones se han realizado con los ficheros entregados desplegados en el servidor de Azure. La página reservas.php tiene dos vistas, no identificado (por defecto) e identificado. Para validar la página identificado se ha copiado el código html que se genera en esa vista y se ha subido a reservID.html.

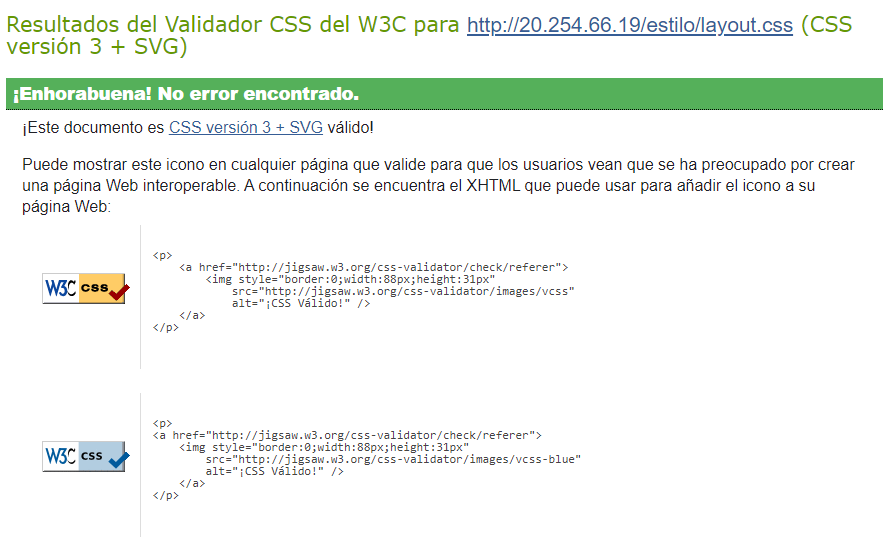








## Usabilidad

Se adjunta en la hoja de cálculo “pruebas\_usabilidad.xlsx” los resultados de los tiempos obtenidos. Para las pruebas se han realizado tres tandas con cuatro usuarios en cada una (doce usuarios en total). En cada tanda, cada usuario ha de realizar tres tareas:

* Tarea 1: Jugar al juego de las preguntas
* Tarea 2: Encontrar el tiempo de la duración de una ruta que sea de su interés
* Tarea 3: Realizar reservas de actividades turísitcas de una semana

Durante las tandas, no se ha guardado simplemente el tiempo de realización de las tareas, si no que se ha observado el comportamiento del usuario y se han escuchado sus sugerencias, quejas o cumplidos. Entre las tandas se han ido realizando distintas mejoras de usabilidad.

### Información previa a considerar

A los usuarios se les ha dispuesto el sitio web durante unos minutos para que se familiaricen antes de realizar las pruebas, pero no todos han profundizado al mismo nivel en él.

En la Tarea 1, algunos usuarios han ido buscando la información por el sitio web para poder responder todas las preguntas bien a la primera, otros simplemente han ido cambiando las respuestas hasta encontrar la correcta y otros no han buscado obtener la puntación máxima.

En la Tarea 3, algunos usuarios han realizado un gran número de reservas distintas para la semana, mientras que otros se han conformado simplemente con realizar unas pocas. De todas formas, esto no ha tenido tanto impacto en los tiempos como lo dicho sobre la Tarea 1.

### Tanda 1

Los usuarios han realizado las tareas en el tiempo previsto teniendo en cuenta el nivel de destreza de cada uno.

En la gráfica comparativa de las plataformas de puede observar que el tiempo en ordenador es siempre mayor que en las otras, esto es debido principalmente a que es en la plataforma en la que realizan la tarea por primera vez.

Los usuarios realizan sugerencias para mejorar la visualización de los formularios y el nombre de las tablas (Tarea 3).

Por esto, se introduce un elemento caption en todas las tablas del sitio web y se crean nuevas reglas de estilo para que los formularios sean más visuales (ahora tienen bordes, los botones son más llamativos y el texto indicando los campos incorrectos es rojo).

### Tanda 2

Esta vez los usuarios han tardado (en general) menos tiempo en realizar las tareas.

De nuevo los usuarios tardan más en la primera plataforma que usan, aún así destaca que tanto en esta como en la anterior tanda los usuarios tardan más usando el móvil que la tablet. Los usuarios dicen que esto se debe a que es dificil distinguir a que pregunta pertenecen los botones de radio (Tarea 1) y que el menú de selección de fecha no es muy cómodo desde el móvil (Tarea 3).

En esta tanda me focalizo en cambiar los botones de radio del juego ya que no encuentro una solución mejor para la selección de fecha desde el móvil. Agrupo las preguntas del juego en fieldsets, aumento su marguen y hago más clara su distinción.

### Tanda 3

Con las tres tandas ya se tiene una perspectiva mayor y se pueden realizar más análisis.

Respecto a la tercera tanda en sí, destacar que el tiempo en móvil se ha reducido en tres de los cuatro usuarios respecto al tiempo en tablet, por lo que se presupone que las mejoras han surtido efecto.

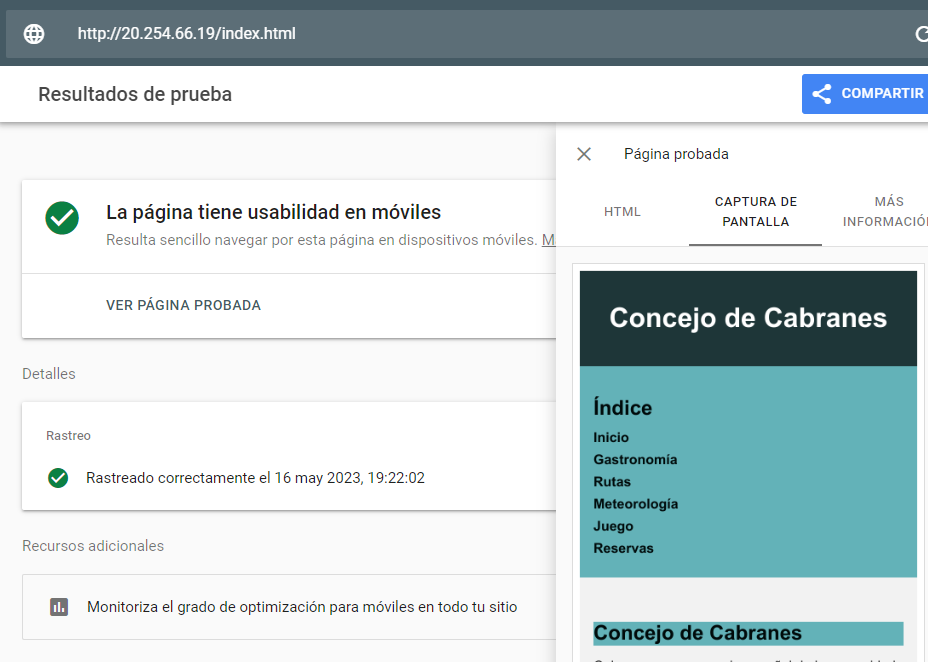
Si nos fijamos solo en el la plataforma de ordenador, vemos que no existe una mejora de tiempo significativa del total. La Tarea 1 ha mejorado levemente, la Tarea 2 se mantiene y de la Tarea 3 poco se puede decir a ciencia cierta, pues ha disminuido y ha aumentado. Puede deberse al nivel de destreza de los usuarios en las distintas tandas, podemos ver lo influyente que puede llegar a ser en el gráfico de la relación destreza-tiempo.

Tomar el total de tiempos de las tareas en cada plataforma puede ayudar focalizar en que parte mejorar la usabilidad. Vemos que los tiempos han empeorado de la segunda a la tercera tanda, pero que realmente el tiempo en móvil es bueno, pues es la primera vez que mejora a la tablet (otro indicio más de que la mejora realizada es útil).

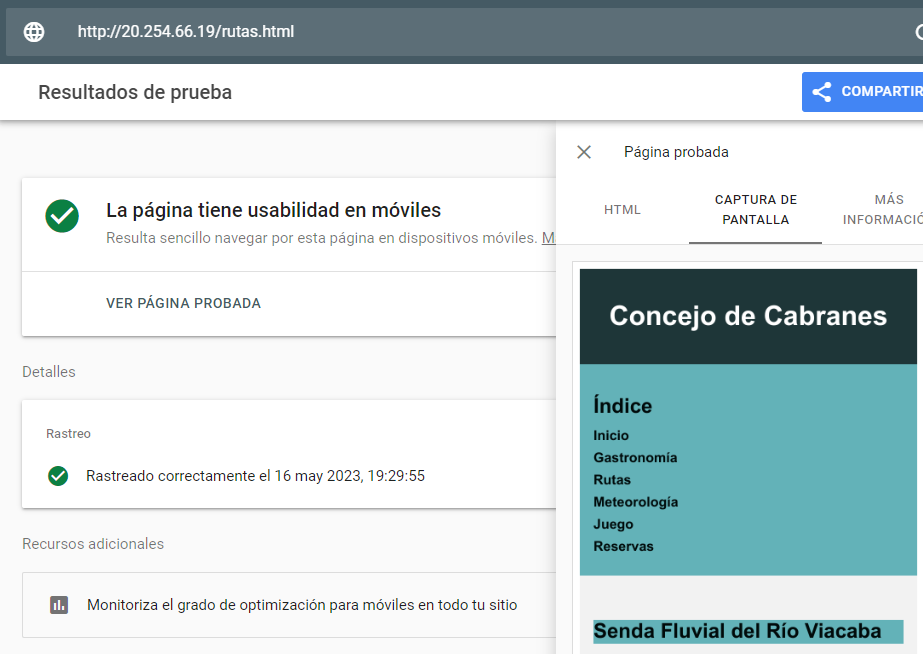
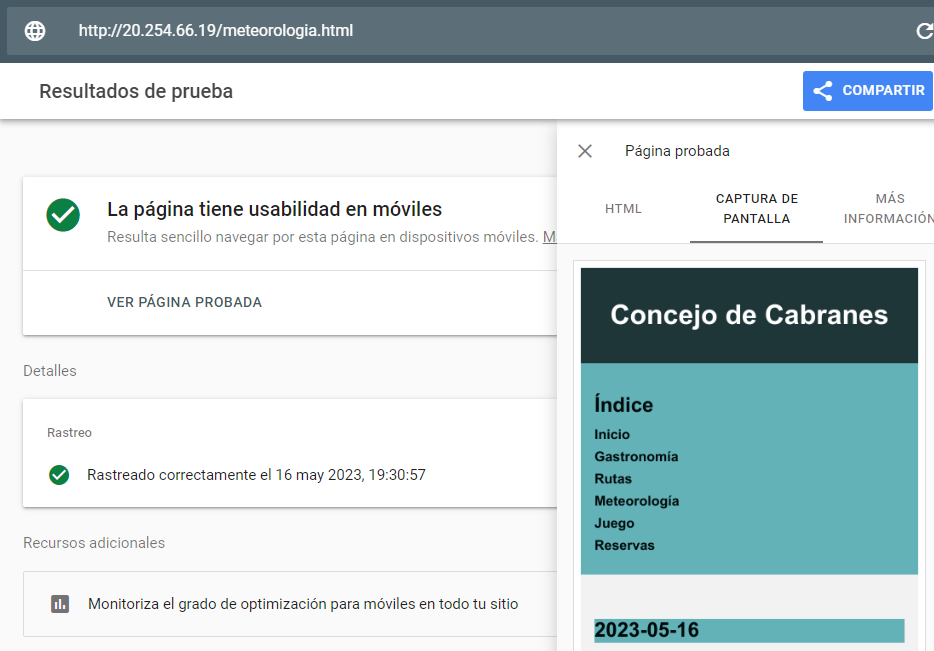
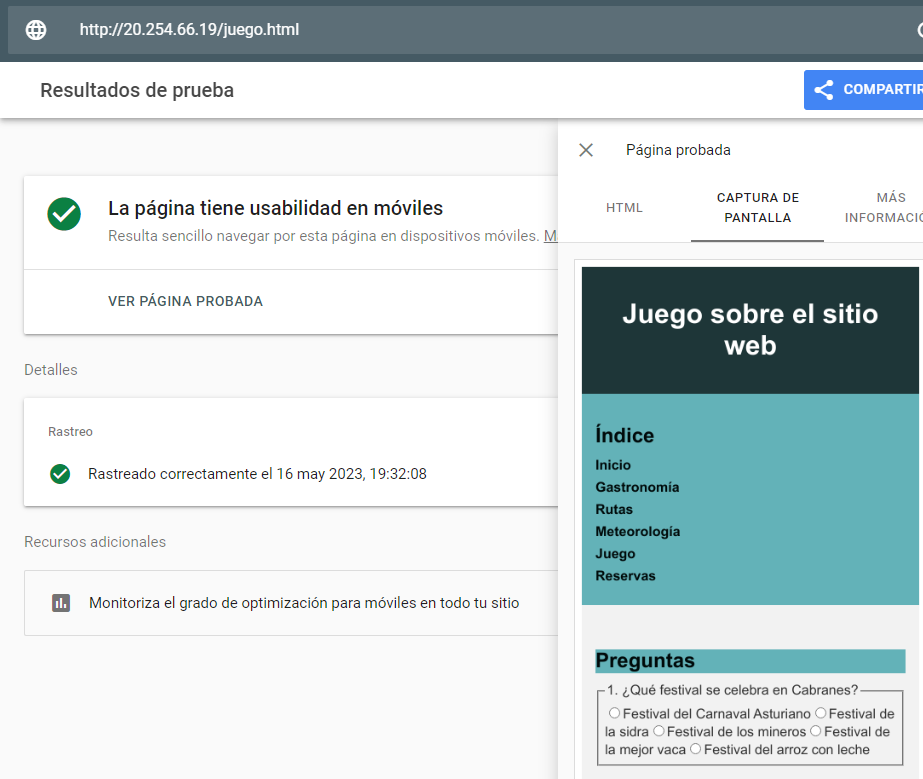
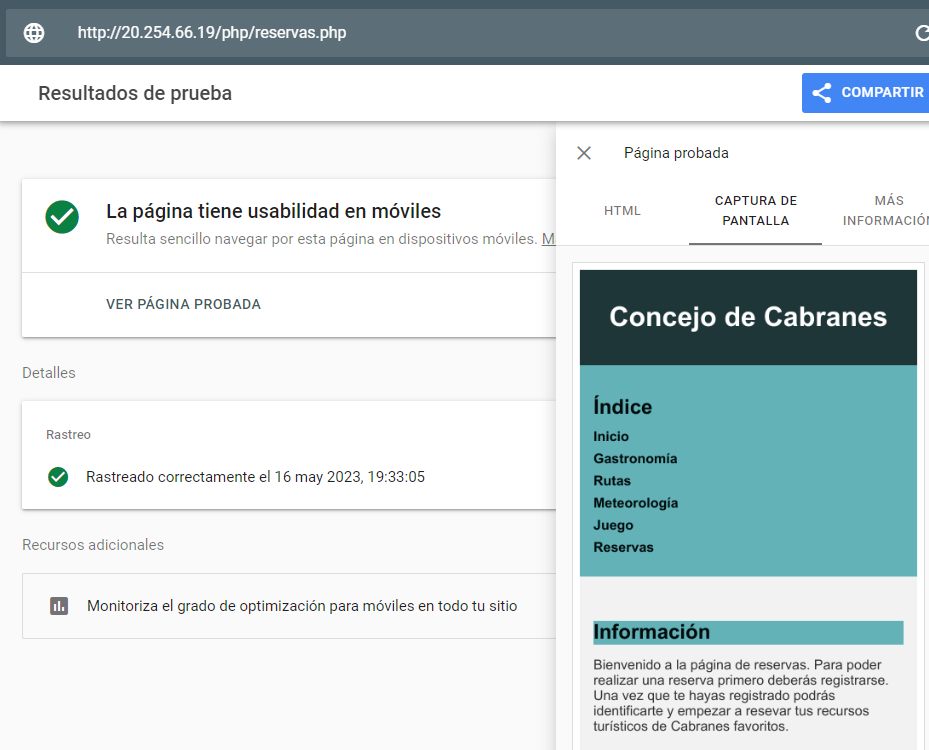
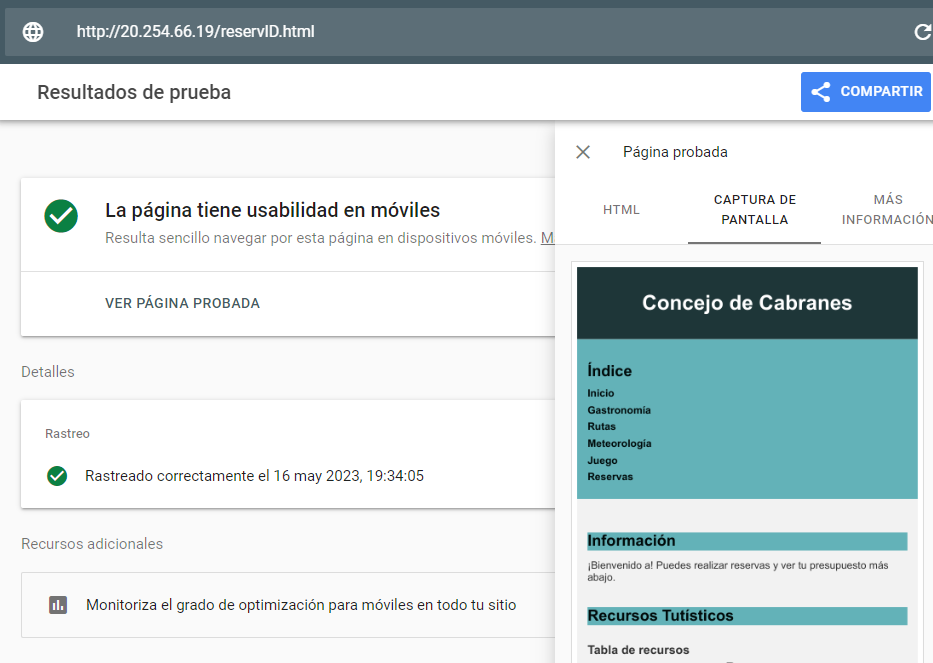
## Adaptabilidad

Todas las pruebas se han realizado con los ficheros entregados desplegados en el servidor de Azure. La página reservas.php tiene dos vistas, no identificado (por defecto) e identificado. Para probar la página identificado se ha copiado el código html que se genera en esa vista y se ha subido a reservID.html.

### Google Mobile-friendly

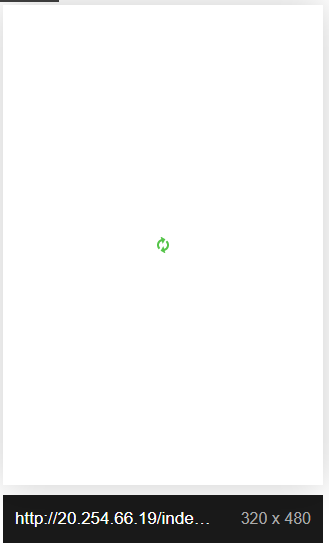


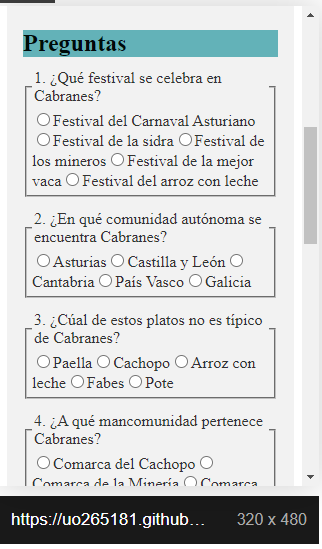
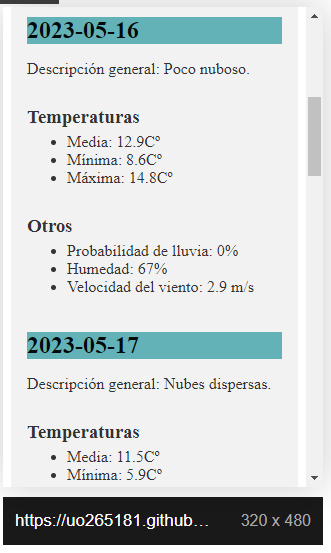
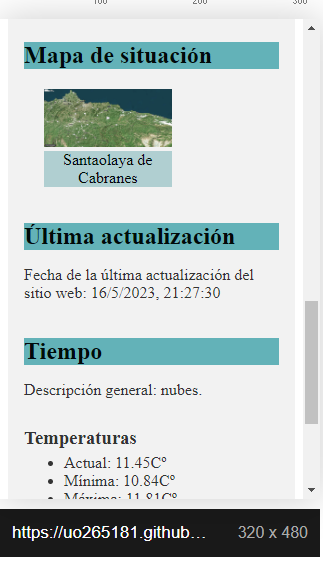
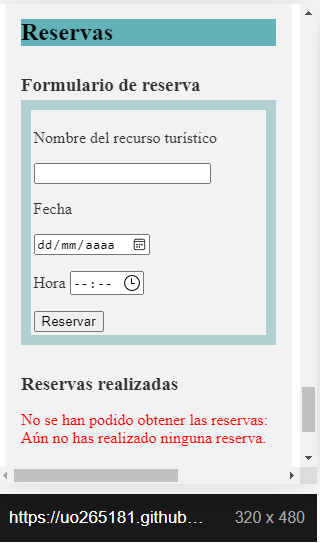


### Screenfly

Debido a que Screenfly no carga las páginas desplegadas en la máquina de Azure (motivo desconozido) se utiliza el despligue realizado en GitHub Pages. Por consecuente, para probar las páginas relacionadas con la parte de reservas, se usan dos html estáticos (reservasEstaticoNoID.html y reservasEstaticoID.html) descargando los códigos html que se generan en dicha parte.



### ResponsiveTestTool

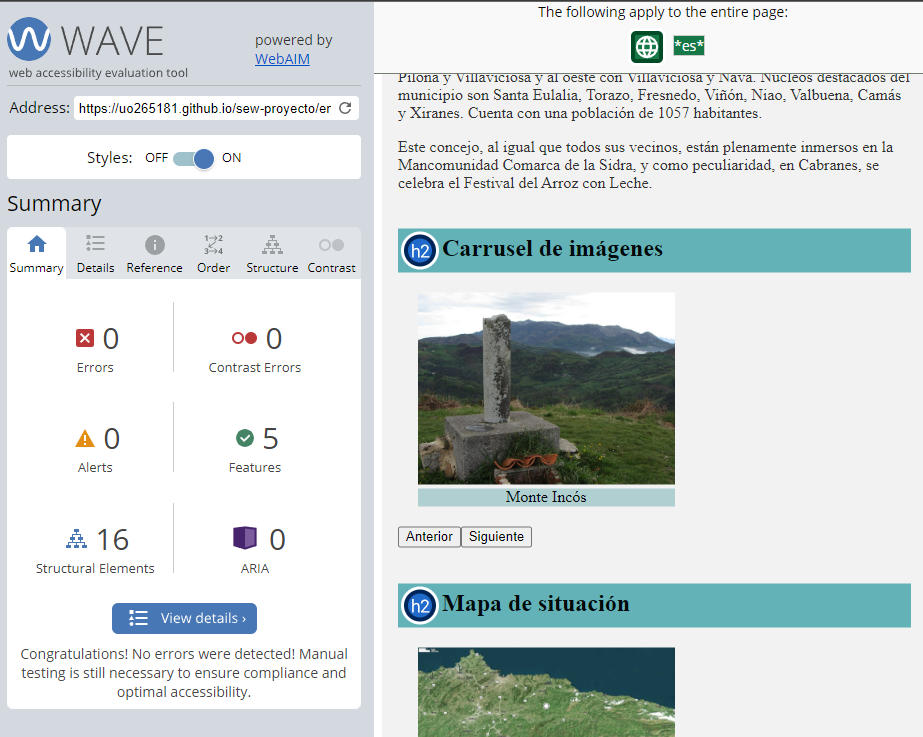
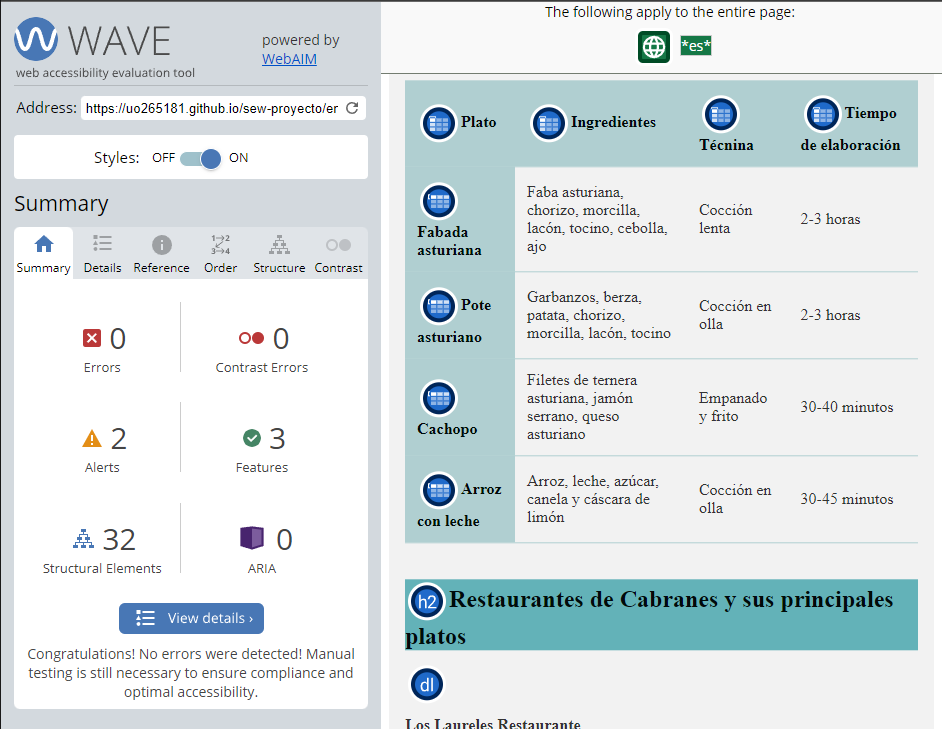
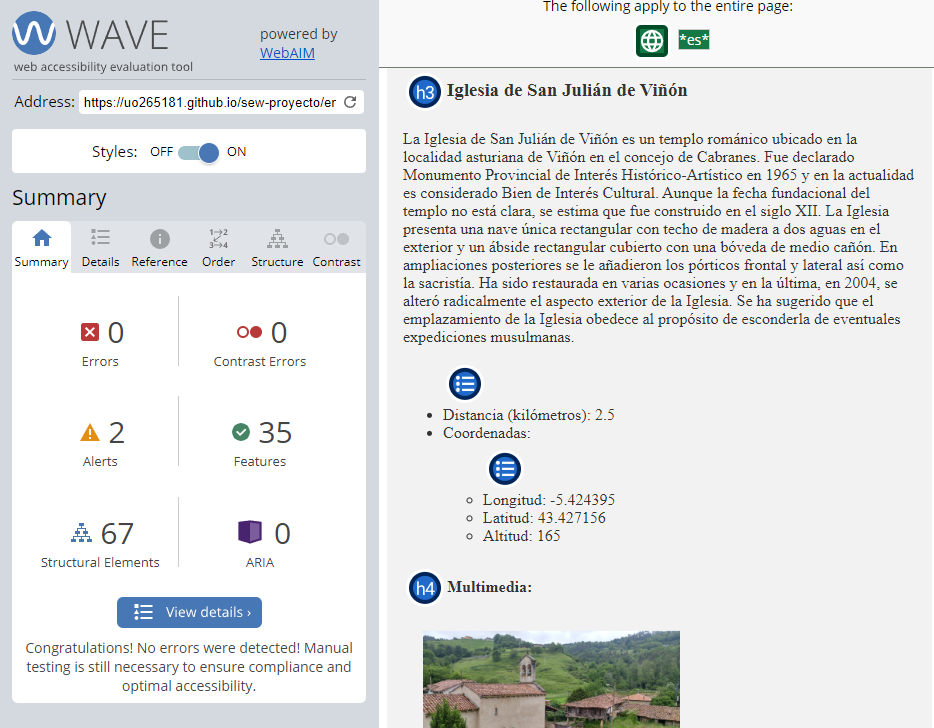
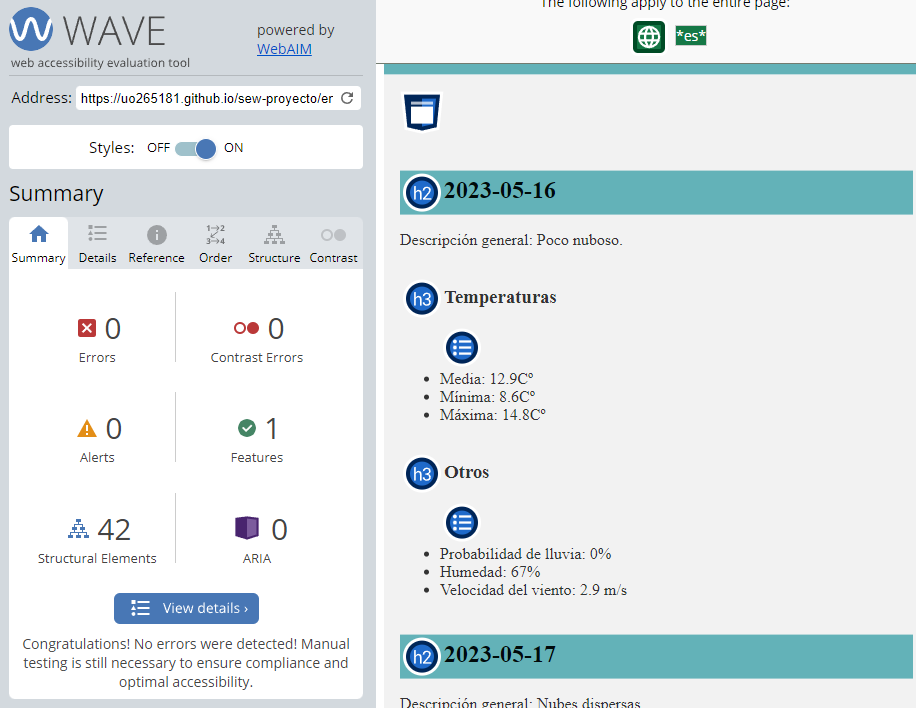
## 

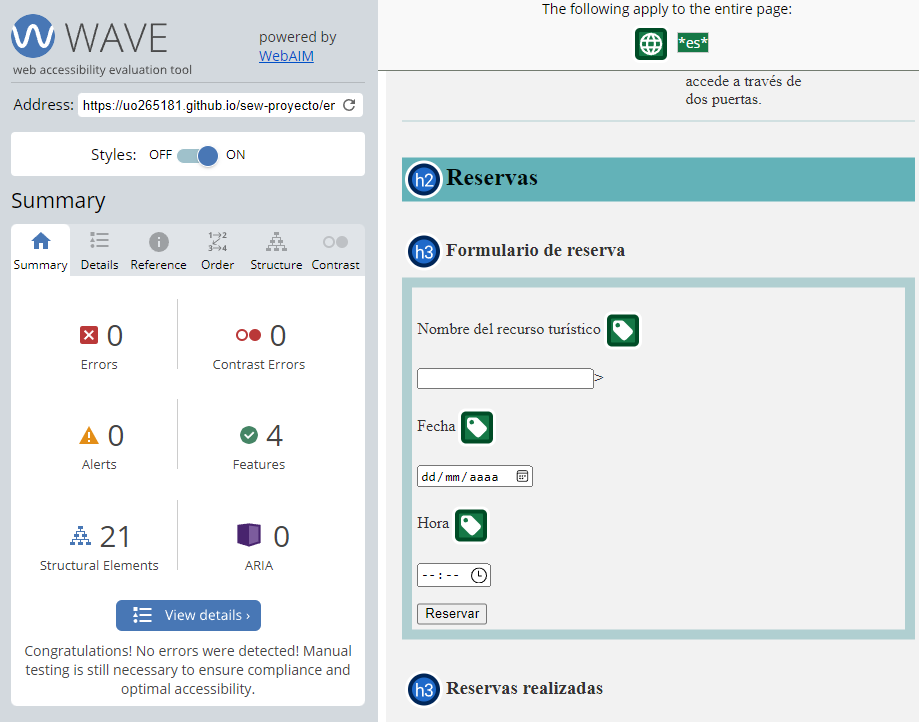
## Accesibilidad

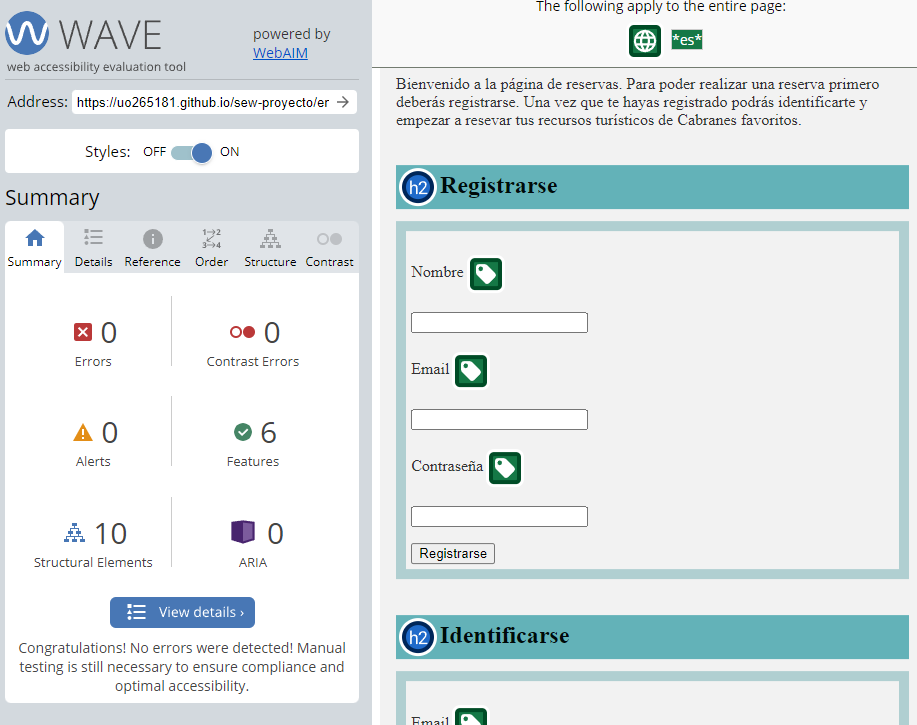
Todas las pruebas se han realizado con los ficheros entregados desplegados en el servidor de Azure. La página reservas.php tiene dos vistas, no identificado (por defecto) e identificado. Para probar la página identificado se ha copiado el código html que se genera en esa vista y se ha subido a reservID.html.

### Wave

La herramienta no procesa el código JavaScript ni aplica las hojas de estilo al sitio web desplegado en Azure (seguramente por seguridad, al ser http), por lo que se prueba con el sitio web desplegado en GitHub Pages. Por consecuente, para probar las páginas relacionadas con la parte de reservas, se crean dos html estáticos (reservasEstaticoNoID.html y reservasEstaticoID.html) descargando los códigos html que se generan en dicha parte.

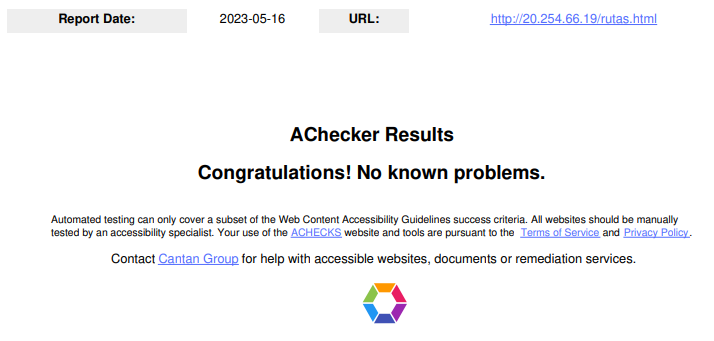
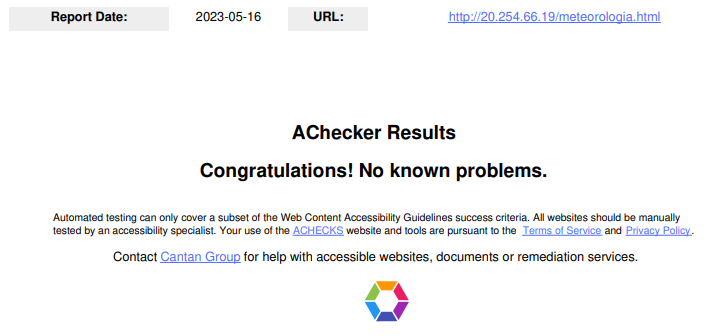
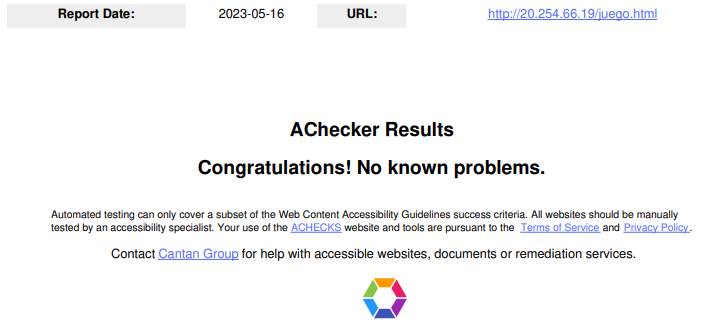


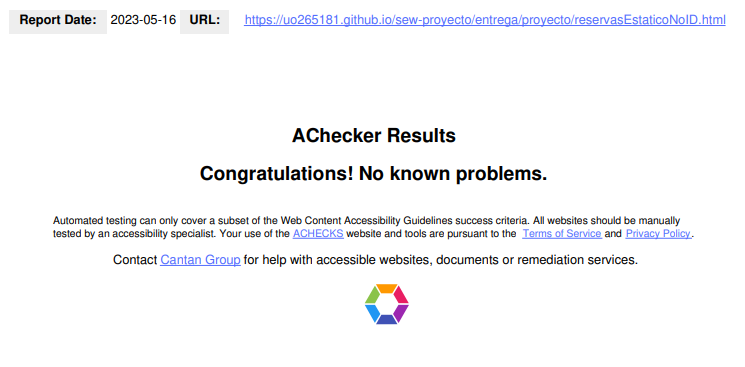
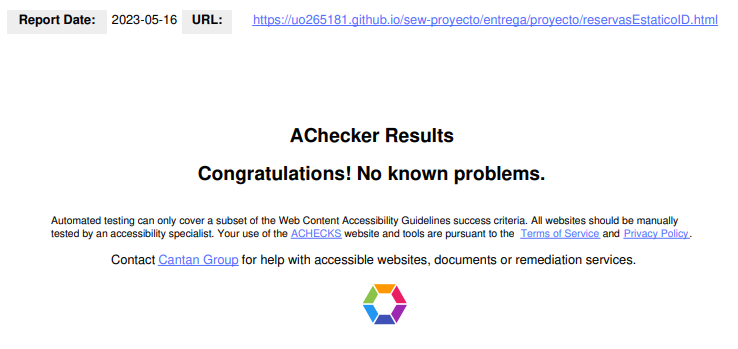


### AChecker

Debido a que es un comprobador de caja negra, para la parte de reservas se utiliza el código estático html generado (reservasEstaticoNoID.html y reservasEstaticoID.html) por si no estuviese generando el código correspondiente.





### Taw

