

# Software y estándares para la Web

---

## **PROYECTO CONVOCATORIA EXTRAORDINARIA 2023/2024**

**VERSIÓN 1.0 (06/02/2024)**

## Contenido

Objetivo .....	2
Temática .....	2
Estructura de archivos del proyecto .....	2
Recursos multimedia del proyecto .....	3
Menú de navegación del proyecto.....	3
HTML .....	3
CSS .....	4
XML.....	5
Computación en el cliente usando ECMAScript.....	6
Computación en el servidor usando PHP .....	7
Bitácora y Pruebas obligatorias a documentar .....	7
Forma de entrega del proyecto.....	9
Anexo 1. Comunidades y ciudades autónomas de España .....	10

## Objetivo

El proyecto de la convocatoria extraordinaria será un proyecto web completo de la temática propuesta (no se admiten ejercicios/módulos independientes) y con la estructura especificada.

Se debe realizar una **Bitácora** explicando todos los pasos del desarrollo y pruebas del proyecto, así como el material consultado (enlaces, videos, libros, etc.). La bitácora deberá contener la fecha y hora de cada entrada. El formato es libre, debe contener obligatoriamente los datos del estudiante. Se presentará en un único archivo en formato PDF.

## Temática

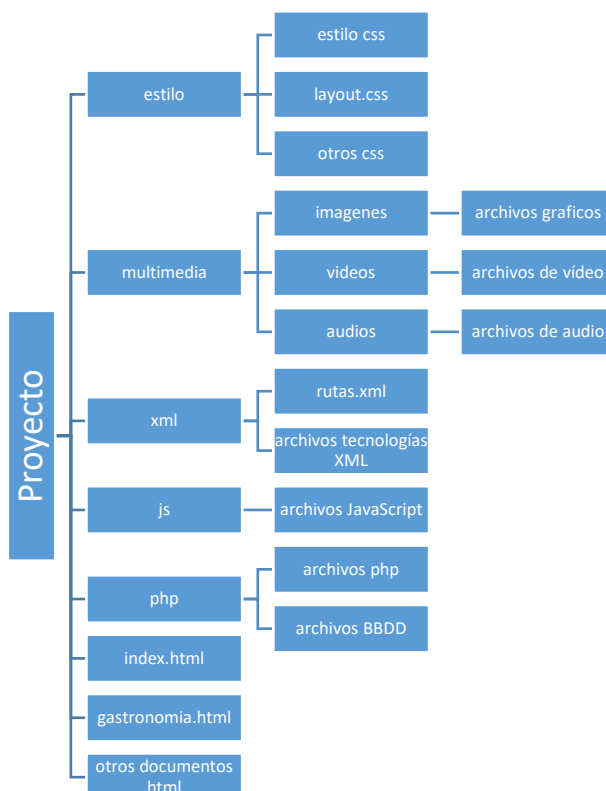
Todos los módulos del proyecto estarán relacionados con la temática.

Se realizará un sitio Web sobre los recursos turísticos de una comunidad o ciudad autónoma de España. El estudiante deberá realizar el proyecto en función de todos los dígitos de su UO con la fórmula siguiente:

**NÚMERO-CCAA = X MÓDULO 19 + 1**

siendo **X** son todos los dígitos del UO y el **NÚMERO-CCAA** es de la clasificación del Anexo 1.

## Estructura de archivos del proyecto



## Recursos multimedia del proyecto

Todos los recursos multimedia del proyecto (imágenes, videos y audios) deben estar ubicados en las carpetas locales definidas en el apartado de “Estructura de archivos del Proyecto”.

**No se permiten enlaces a imágenes, vídeos o audios de sitios web externos o almacenados en onedrive u otros sistemas en la nube**

## Menú de navegación del proyecto

El sitio web del proyecto debe tener un menú de navegación estático que muestre todas las opciones para el acceso a los recursos turísticos de la comunidad o ciudad autónoma. El menú de navegación debe aparecer en todas las páginas del proyecto.

El contenido debe ser:

- Página principal
- Gastronomía
- Rutas
- Meteorología
- Juego
- Reservas

## HTML

Documento HTML5 **completamente estático**. Debe utilizarse el marcado semántico correcto, así como la estructuración correcta de los contenidos.

Se debe crear un documento denominado “**gastronomia.html**” que será enlazado desde la opción del menú de navegación “Gastronomía”.

Este documento debe mostrar información relativa a la gastronomía típica de la comunidad o ciudad autónoma.

Debe tener **al menos** los siguientes elementos HTML:

- Lista ordenada con al menos 3 elementos
- Lista no ordenada con al menos 3 elementos
- Lista de definición con al menos 3 elementos
- Una de las listas debe tener listas anidadas (pudiendo ser de varios tipos de lista)
- Tabla accesible con al menos 2 columnas y 4 filas de datos. Se requiere el uso de IDs para hacer la tabla accesible.
- Recursos multimedia:
  - Al menos una imagen adaptable y accesible
  - Al menos un vídeo adaptable y accesible servido con al menos 2 formatos diferentes. No es obligatorio poner subtítulos
  - Al menos un audio adaptable y accesible servido con al menos 2 formatos diferentes. No es obligatorio poner subtítulos

- Utilización de elementos semánticos de HTML para dotar de estructura semántica a los contenidos.

## CSS

Las hojas de estilo deben ser aplicadas a todos los documentos que componen el sitio web.

La definición de selectores debe ser correcta y óptima, está prohibido el uso de ID y class como selectores de estilo.

Todos los selectores de las hojas de estilo deben ir precedidos por su especificidad como un comentario.

Se deben añadir comentarios en las reglas de las hojas de estilo, justificando las advertencias del validador de CSS del W3C, en concreto, la herencia de colores (indicando de donde se hereda) y las redefiniciones de las propiedades.

Se debe garantizar la adaptabilidad del sitio web.

Las hojas de estilo deben ser:

- **estilo.css**: debe contener las características generales de estilo que se aplican a todos los documentos del sitio web. Contendrá los estilos relativos a:

- Tipos de fuentes
- Propiedades de colores de los elementos
- Propiedades de aspecto de los enlaces
- Propiedades de listas y tablas
- Recursos multimedia

Se debe tener en cuenta la herencia de las propiedades para realizar una correcta definición de la hoja de estilo.

- **layout.css** con las propiedades necesarias para la implementación de la maquetación de todos los documentos que componen el sitio web. Solo se permite el uso de los módulos de CSS

- [CSS-Flexible Box Layout]
- [CSS-Grid layout]
- [CSS- CSS Multi-column Layout]

Opcionalmente se permite la creación de hojas de estilo adicionales que deben ser justificadas en la Bitácora.

## XML

Debe diseñarse un archivo XML, de complejidad similar a las prácticas de la convocatoria ordinaria.

- Deberá contener validadores DTD y Schema
- Deberá traducirse a HTML usando ECMAScript y jQuery, que será enlazado desde la opción del menú de navegación “rutas”.
- Deberá traducirse al formato KML usando ECMAScript y jQuery, que será utilizado para mostrar la planimetría de las rutas turísticas.
- Deberá traducirse al formato SVG usando ECMAScript y jQuery, que será utilizado para mostrar el perfil de altimetría de las rutas turísticas.
- No está permitido la utilización de transformaciones XSLT por problemas de seguridad.
- No está permitido la utilización de lenguajes diferentes de ECMAScript.

Se debe construir un archivo XML (**rutas.xml**) con 3 o más rutas por la comunidad o ciudad autónoma a pie, en bicicleta u otro medio de transporte con paradas en los principales atractivos turísticos.

El archivo XML deberá tener los siguientes elementos o atributos:

- Nombre de la ruta turística (por ejemplo “Ruta de los molinos”)
- Tipo de ruta (por ejemplo “Arquitectura y monumentos”, “Gastronómica”, “Paisajística”, “Mixta tapas y monumentos”, “Escalada”, “Senderismo”, etc.)
- Medio de transporte (por ejemplo “A pie”, “Bicicleta”, etc)
- Fecha de inicio de la ruta (elemento o atributo marcado como opcional)
- Hora de inicio de la ruta (elemento o atributo marcado como opcional)
- Tiempo de duración de la ruta (por ejemplo “2 horas”, “3 días”, “2 semanas”)
- Agencia que gestiona la ruta (por ejemplo “Sin agencia”, “NaturaTur”)
- Descripción de la ruta (elemento que contiene texto)
- Personas adecuadas para la ruta (por ejemplo “Se puede ir con niños”, “Personas en buena forma física”, “tercera edad”, etc.)
- Lugar de inicio de la ruta (por ejemplo “Plaza del ayuntamiento”)
- Dirección de inicio de la ruta (por ejemplo “calle mayor”)
- Coordenadas geográficas de inicio de la ruta: longitud, latitud y altitud
- Referencias y bibliografía con información de la ruta (mínimo 3)
  - Referencia 1: por ejemplo <https://es.wikipedia.org/wiki/Foncalada>
  - Referencia 2: <http://prerromancoasturiano.es/>
  - Referencia 3. etc.
- Recomendación de la ruta de 0 a 10 (por ejemplo “7”)

- Hitos de la ruta con algún tipo de atractivo turístico (mínimo 3 hitos):
  - Nombre del hito
  - Descripción del hito
  - Coordenadas geográficas del hito: longitud, latitud, altitud
  - Distancia desde el hito anterior (las unidades se expresarán como atributos)
  - Galería de fotografías del hito (mínimo 1, máximo 5)
    - Fotografía 1: Por ejemplo Monumento.jpg
    - Fotografía 2: Por ejemplo Panorama.jpg
    - Fotografía 3: etc...
  - Galería de vídeos del hito (opcional). Mínimo 0 y máximo 3.
    - Video 1: Por ejemplo Paisaje360.mpeg o enlace a YouTube, Vimeo, etc.
    - Video 2: Por ejemplo Modelo3D.mpeg
    - Video 3: etc.
- Planimetría de la ruta usando el archivo KML generado.
- Altimetría de la ruta usando el archivo SVG generado.

## Computación en el cliente usando ECMAScript

- Deberá usarse obligatoriamente el paradigma de orientación a objetos en ECMAScript. No se admitirá el paradigma procedimental ni otros paradigmas soportados por ECMAScript.
- No se pueden utilizar bibliotecas externas (a excepción de jQuery)
- Deberá usarse jQuery obligatoriamente encapsulado dentro clases y métodos para mantener el paradigma de orientación a objetos.

La **página principal** del proyecto, **index.html**, deberá utilizar ECMAScript para:

- Crear un carrusel de un mínimo de 5 fotos con los principales recursos turísticos de la comunidad o ciudad autónoma usando ECMAScript puro.
- Mostrar la información meteorológica en tiempo real relativa a la capital de la comunidad autónoma o a la ciudad autónoma consumiendo Servicios Web.
- Visualizar el mapa de situación de la comunidad o ciudad autónoma usando una imagen.
- Incorporar una sección de noticias sobre la comunidad o ciudad autónoma consumiendo Servicios Web, indicando la fecha y hora de la última actualización de las noticias usando ECMAScript puro.
- Indicar la fecha y hora de la última actualización del documento index.html usando ECMAScript puro.

En la opción de “**Meteorología**” del menú de navegación deberá consumirse Servicios Web para mostrar las previsiones meteorológicas en la comunidad o ciudad autónoma para los próximos 7 días.

En la opción de “**Juego**” del menú de navegación deberá desarrollarse en ECMAScript puro un juego de 10 preguntas de tipo test con el objetivo de conocer la experiencia de usuario con el proyecto desarrollado. Cada pregunta con 5 opciones de respuesta y una sola respuesta correcta. Las preguntas versarán únicamente sobre información contenida en el sitio web del proyecto. Se obtendrá una calificación de 0 a 10 en función del número de respuestas acertadas (1 punto) o falladas (0 puntos). Es obligatorio que el jugador responda todas las preguntas.

## Computación en el servidor usando PHP

La computación en el servidor se utilizará para la creación y gestión de una Base de Datos usando obligatoriamente MySQL o MariaDB.

La base de datos debe tener un mínimo de 5 tablas relacionadas y normalizadas, y deberá entregarse el diagrama E-R en la bitácora.

- Deberá usarse obligatoriamente el paradigma de orientación a objetos en PHP. No se admitirá el paradigma procedimental ni otros paradigmas soportados por PHP.
- Deberá usarse el sistema de gestión de bases de datos en el servidor
- No se pueden usar bibliotecas externas

**El usuario y contraseña de la Base de Datos MySQL o MariaDB debe especificarse en la bitácora.**

El objetivo será simular una central de reservas de los recursos turísticos. Los usuarios deben:

- Registrarse como usuario mediante un formulario.
- Antes de realizar una reserva es necesario identificarse como usuario registrado.
- Para realizar una reserva se debe:
  - Seleccionar las fechas y horas del recurso turístico a visitar.
  - Cada recurso debe tener:
    - Límites de ocupación y la aplicación deberá comprobar dichos límites.
    - Precios de cada recurso turístico.
    - Breve descripción de cada recurso turístico.
  - Se generará un presupuesto de todos los recursos turísticos reservados por el usuario

Los recursos turísticos pueden ser de diferentes tipos: museos, rutas, restaurantes, hoteles, etc. Es necesario tener un mínimo de 10 recursos turísticos.

## Bitácora y Pruebas obligatorias a documentar

Se debe realizar una Bitácora explicando **todos los pasos del desarrollo y pruebas del proyecto**, así como el material consultado (enlaces, videos, libros, etc.).

La Bitácora deberá contener la fecha y hora de cada entrada. El formato es libre, debe contener obligatoriamente los datos del estudiante. Se presentará en un único archivo en formato PDF.



En la Bitácora se documentarán además las pruebas siguientes:

- Funcionamiento en máquina local mediante XAMPP
- Funcionamiento en la nube (Azure) con una cuenta de estudiante
- Validación del cumplimiento de los estándares HTML5 y CSS3, **sin errores ni advertencias**, mediante los validadores del W3C (tanto del código estático como el código generado).
- Pruebas de usabilidad del proyecto
  - Al menos 12 personas indicando su edad, género y nivel de destreza en el uso de ordenador (de 0 a 10), en 3 tandas de 4 personas.
  - Se deben guardar los tiempos (en segundos) que tarda cada usuario en realizar cada tarea.
  - Se deben documentar las reacciones y comentarios de cada usuario en el uso del proyecto.
  - Entre las tandas se debe especificar en la Bitácora los cambios realizados para mejorar la usabilidad.
  - Se deben repetir las pruebas en ordenador, tableta y móvil
  - Los resultados de las pruebas de usabilidad se guardarán en una hoja de cálculo, una fila por usuario y en columnas cada una de las pruebas.
  - Con los datos de la hoja de cálculo se harán estadísticas y gráficas con el objetivo de observar la mejora de la usabilidad según avanzan las tandas que se documentará en la Bitácora.
  - Las tareas a realizar por cada persona son:
    - TAREA1: Jugar al juego desarrollado.
    - TAREA2: El usuario debe encontrar la información sobre el tiempo de duración de la ruta turística de su interés.
    - TAREA3: Hacer una reserva de actividades turísticas durante una semana y obtener el presupuesto.
- Pruebas de adaptabilidad usando tres validadores y garantizando la adaptabilidad.
- Pruebas de accesibilidad utilizando tres herramientas, garantizando no tener errores de accesibilidad de “verificación automática”. El nivel exigido es AAA.

## Forma de entrega del proyecto

Se deben entregar **3 archivos cada uno en una tarea en el campus virtual**:

- **“UOXXXXX-proyecto.zip”** con el contenido completo del proyecto desarrollado en la carpeta “Proyecto” con la **estructura especificada** en el apartado “Estructura de archivos del proyecto”:
  - Todos los documentos HTML
  - Todas las hojas de estilo CSS
  - Todos los recursos multimedia (imágenes, videos y audios)
  - Todos los archivos de tecnologías xml
  - Todos los archivos js
  - Todos los archivos php
- **“UOXXXXX-Bitácora.pdf”** con el contenido en formato PDF de la Bitácora realizada.
- **“UOXXXXX-Usabilidad.xlsx”** con el contenido de la hoja de cálculo creada en el desarrollo de las pruebas de usabilidad.

Consideraciones importantes:

- Tamaño máximo de cada entrega en el campus virtual es de **90MB**. Es responsabilidad del estudiante que el contenido de las entregas se adapte a ese tamaño.
- Todos los recursos multimedia utilizados deben estar en la carpeta Proyecto
- No se admiten enlaces a onedrive u otros sistemas en la nube.

Se considerarán como “no entregados”:

- los proyectos que no cumplan el formato de entrega y el nombrado de los archivos y el contenido.
- los archivos que estén protegidos con contraseña.

**Se deben tener en consideración las pautas y normas de la asignatura, el no cumplimiento de las mismas invalidará el proyecto presentado.**

## Anexo 1. Comunidades y ciudades autónomas de España

El listado contiene los 19 comunidades y ciudades autónomas de España

	Nombre	Capital
1	Aragón	Zaragoza
2	País Vasco	Vitoria
3	Castilla y León	Valladolid
4	Comunidad Valenciana	Valencia
5	Castilla-La Mancha	Toledo
6	Andalucía	Sevilla
7	Galicia	Santiago de Compostela
8	Cantabria	Santander
9	Comunidad Foral de Navarra	Pamplona
10	Islas Baleares	Palma
11	Principado de Asturias	Oviedo
12	Región de Murcia	Murcia
13	Extremadura	Mérida
14	Melilla	Melilla
15	Comunidad de Madrid	Madrid
16	La Rioja	Logroño
17	Canarias	Las Palmas de Gran Canaria
18	Ceuta	Ceuta
19	Cataluña	Barcelona