

# Especificaciones proxecto multimódulo

2º CS Desenvolvemento de Aplicacións Web

Curso 2021 - 2022

Versión 1.0

4 de decembro de 2021

## Requerimentos DIW

### 1. Planificación

Para realizar una buena planificación del proyecto debes dar una breve descripción del proyecto. A la hora de realizar la maquetación Web, deberemos pensar previamente:

1. Cuáles son los elementos que va a contener cada una de nuestras páginas.
2. Cómo irán colocados cada uno de esos elementos.
3. Realizar una serie de bocetos preliminares de lo que será nuestro sitio Web.
  - Un boceto debe reflejar la interactividad y la funcionalidad del sitio Web.
  - Se debe hacer una distribución de los grandes bloques de elementos de información .
4. Todas las páginas del sitio deben mantener una misma estructura. Las opciones de navegación deben ser claras:
  - El número de clics para acceder a los contenidos debe minimizarse
  - El usuario debe saber dónde está
  - Imagen corporativa de todas las páginas del sitio Web

### 2. Responde a las siguientes cuestiones

1. **¿A quién va dirigido?**
  - Perfil demográfico objetivo:
  - Idioma/idiomas
2. **Competencia directa:** Consulta algunas páginas similares para
  - Cómo se financia, en caso de que obtenga beneficios
  - Puntos fuertes y débiles
  - Redes sociales con las que se complementa (si las tiene y si tu página las necesita)
3. **Sentimientos y sensaciones que quieres transmitir. Razonar**
  - **Color:** Debes incluir la paleta de color y el contraste entre ellas.
  - Tipografía
4. **Mapa de navegación:** Debes indicar las opciones del menú de navegación (mínimo 3 opciones y tipo de mapa de navegación).

5. **Imágenes:** Debes incluir formato adecuado de las imágenes para tu página, y que dichas imágenes no tengan copyright.
3. **Entregar:**
1. Maqueta página en aplicación Figma o similar. Debe incluir:
    - Diseño aplicación pantalla general
    - Diseño dispositivo móvil
    - Diseño logotipo acorde con el tema de la página
  2. Requisitos mínimos:
    - El diseño debe ser actual
    - Visual, debes incluir imágenes o algún elemento multimedia con formato adecuado
    - Implementar un criterio de accesibilidad, en función del público objetivo
3. **Documentación**
- Breve descripción del proyecto
  - Planificación del proyecto
  - Criterios de accesibilidad, y usabilidad
  - Cookies, copyright y protección de datos
  - Formato documentación acorde al diseño de la página
  - Formato documento pdf o html
4. **Implementación**
- Debes utilizar lenguaje HTML5 y CSS(3). Se valorará positivamente el lenguaje sass
  - La página debe ser responsive y adaptable. Para ello debes utilizar Flexbox o gridCSS, (también puedes combinar ambos), y mediaqueries. La página debe poder ser visualizada en pantallas con distintas resoluciones y tamaños.
  - Contenido acorde al sitio Web con mensajes claros y concisos
  - Tener en cuenta criterios de usabilidad (acceso y manejo simple y sencillo), y criterios de accesibilidad (mínimo uno)
  - Derechos de autor, copyright, etc.
  - Estructura de carpetas.

## Requerimientos DWCC

Para o módulo de DWCC deberá deseñarse.

Polo menos unha pantalla que lea datos dun ficheiro JSON xerado desde o código PHP, amosar eses datos na pantalla, presentándoos nunha páxina con diferentes contedores e ben organizados, seguindo as directrices do deseño de DIW. Sería aconsellable que entre os datos haxa imaxes.

Os elementos JSON, deben organizarse internamente como un array de obxectos que teña varios campos.

Se a web ten un formulario de rexistro, o nome do usuario rexistrado, debe gardarse nun obxecto tipo Storage, de xeito que apareza nalgunha parte da páxina, o nome do usuario que está conectado. Debe recuperarse a sesión, en caso de que se peche o navegador accidentalmente, unha vez se conecta ao dominio de novo. E debe borrarse en caso de que o usuario peche a sesión.

O formulario de rexistro, ou calquera outro que se formule no proxecto, debe levar validación de campos con Javascript. Emitindo unha mensaxe identificando o erro e alertando do mesmo ao usuario, ou en caso de que os datos sexan correctos, debe amosarlle ao usuario que os datos son correctos.

Para a recuperación dos elementos JSON deben de empregarse funcións asíncronas, empregando os métodos Asyn - Await, promises e a API Fetch.

Debe empregarse unha API de xeolocalización, polo menos para localizar unha dirección física nun mapa. A dirección pode ser por exemplo, a da empresa que realiza a Web, ou a que se considere pero é coñecida de antemán.

Aspectos técnicos:

- Debe definirse unha clase ES6 cos datos que se recollan do JSON.
- Traballar con arrays e cos métodos de arrays de javascript.
- Cada un dos elementos JSON, será unha instancia a un obxecto desa clase definida.
- Para a carga dos datos JSON deben aplicarse comunicación asíncrona e a API Fetch. E os mecanismos de detección e presentación de erros Try-Catch.
- Emprego dunha API de xeolocalización.

Aspectos de programación:

- Debe estar o código organizado por MODELO-VISTA-CONTROLADOR. Nun ficheiro ou en varios. Pero que se vexan claramente as tres partes.
- Deben estar comentadas as variables globais, e explicadas para que son.
- Cada función debe ter un comentario breve de que é o que fai, que parámetros recibe e que devolve.
- O programa farase o máis modular posible, tentando refactorizar todo aquel código que vexamos se repita.

Valorarase (ademais dos CA correspondentes):

1. Que o programa funcione
2. Que estén ben divididas as partes de Modelo-Vista-Controlador.
3. O emprego de métodos de arrays propios de Javascript. Que optimicen o funcionamento do programa e que substitúan a bucles (for, while) innecesarios.
4. A refactorización de código.
5. Que as variables globais estean ben definidas, e sexan as que deben ser e non máis. Que estea xustificando o uso como variable global.

6. Captura dos erros derivados da comunicación asíncrona.
7. A limpeza de código e o uso de comentarios.

## Requerimentos DWCS

Como el objetivo principal del módulo es el dominio del lenguaje PHP, la evaluación del trabajo se realizará sobre el código PHP generado, la conexión con una base de datos relacional (a ser posible normalizada) y haber cumplido con las especificaciones que más adelante se detallan.

### Requisitos funcionales.

A la aplicación web podrán acceder 3 tipos de usuarios:

- Usuarios anónimos (tipo3). Un usuario anónimo es aquel usuario que no se haya identificado en la aplicación web y no posea una sesión activa.
- Usuarios registrados (tipo2). Un usuario registrado es aquel que haya realizado el registro de sus datos, al menos con el nombre completo, correo electrónico, teléfono, dirección completa y contraseña, y se identifique en la aplicación.
- Usuarios administradores (tipo1). Un usuario administrador tiene permisos de administración y puede ser añadido por otro usuario administrador.

Operaciones:

1. **Identificación:** Operación que realiza un usuario para poder acceder a la aplicación en función de su perfil. Una vez identificado, la aplicación deberá guardar en la base de datos el momento del acceso.
2. **Alta del usuario.** Debemos registrar el usuario en el sistema, recogiendo sus datos por medio de un formulario. El usuario puede subir ficheros relacionados con su perfil (por ejemplo, foto, cv. , etc.) y que se guardarán en una carpeta en el disco duro asociada a su identificador de usuario.
3. **Modificación del perfil:** Cualquier usuario registrado puede modificar los datos asociados a él mismo, registrando la fecha de forma automática.
4. **Búsqueda de información:** Operación que realizan los 3 tipos de usuarios. Debe de trabajar con fechas (por ejemplo, inicio y final) y la información presentada debe diferir en su cantidad en función del perfil. A los usuarios anónimos sólo se les mostrará una información muy básica que les permita curiosar en la web.
5. **Vista detallada de algún elemento de la aplicación que permita ser reservado y un calendario de disponibilidad.** Operación que realizan los usuarios identificados a partir de la búsqueda de algún campo (a decidir por el programador). En esta sección deberá tener algún carrusel de imágenes.
6. **Solicitud de reserva de algún elemento de la aplicación:** Operación que realizan los usuarios identificados entre dos fechas. Una vez se solicita la reserva se deberá

guardar el momento de la reserva y enviar un email al administrador indicando que tiene pendiente la aceptación de una reserva junto con toda la información.

7. **Histórico:** Ver los datos de las modificaciones del perfil entre unas fechas dadas. Cada usuario ve las suyas y el administrador ve todas, pudiendo filtrar por algún campo.
8. **Gestión (backend):** Operaciones realizadas por un usuario administrador que se implementará con una interfaz de tipo:
  - Añadir / Eliminar elementos de las aplicaciones. Para el caso del borrado se debe preguntar al usuario si está de acuerdo (especificando alguna dependencia en el caso de que la hubiese) y borrando todos los elementos en una operación idempotente.
  - Modificar características de algún elemento de la aplicación.
  - Aceptación de la realización de la reserva. Para ello el sistema le mostrará un listado de los elementos pendientes de ser reservados. Tanto como si se acepta o no se acepta la reserva, se debe enviar al cliente un email informando de cómo ha quedado su solicitud.

### **Requisitos técnicos.**

1. Desarrollo en PHP 7/8 bajo el paradigma de programación orientada a objetos (POO). Dentro de este paradigma se pide:
  - Manejo de excepciones en PHP.
  - Empleo de transacciones
  - Implementar herencia
  - Las clases deben estar en ficheros separados, definir namespaces y se deben cargar automáticamente
  - Utilizar Composer con PHPmailer para el envío de correos
  - Las passwords se deben guardar en la BBDD codificada
  - Utilización de cookies y sesiones
  - Empleo de polimorfismo en la búsqueda
  - Guardar los datos de la conexión a la BBDD en un fichero XML independiente, fuera de la raíz de documentos y validarlo con un esquema XSD
2. Utilización de la interfaz de acceso a base de datos mediante objetos y utilización de sentencias preparadas para evitar inyección SQL
3. Base de datos relacional MySQL con datos de ejemplo para su prueba, pero el acceso a los mismos es a través de PDO.
4. Se valorará la separación de la parte gráfica del procesado siguiendo un paradigma MVC. En todo caso el envío de información entre cliente y servidor se debe hacer a través de JSON.

### **Otras consideraciones.**

- Debemos poblar la base de datos con al menos 10 filas por tabla. Debe haber una tabla donde se guarde el nombre de los ficheros y su ubicación en el disco duro.

- Hay que subir la aplicación a un alojamiento web y poder conectarse a ella.

### **Documentación.**

Debemos especificar claramente lo que va a realizar nuestra aplicación y las funcionalidades implementadas. Además, habrá que proporcionar el código fuente en un fichero comprimido.

## **Requerimentos Despregamento de AW**

- Crear a documentación do código PHP con PHP Documentor ou Doxygen.
- Ir subindo ao longo do desenvolvemento as versións do proxecto a GitHub o similar (compartir o repositorio co usuario mrey-profe). Debe estar subido todo o código necesario para despregar o proxecto, incluída a base de datos e os scripts para a súa xeración.
- Despregar a aplicación en Dinahosting e nalgún servizo na nube.
- Crear un sitio virtual SEGURO, con certificado SSL.
- Configuración do DirectoryIndex, personalización (mediante arquivos web, non mensaxes de texto) do erro 404, eliminación da indexación do directorio.

## **Outras consideracións**

- Tanto para a entrega de documentación como para o propio sitio web deberase coidar a presentación, o correcto uso de signos de puntuación (e espazos ao redor deles), a ortografía e a gramática.
- Esta documentación pode sufrir pequenas modificacións coas aclaracións que vaian xurdindo ao longo do desenvolvemento do mesmo.