

## Software y estándares para la Web

---

### **PROPUESTA DE PROYECTO CONVOCATORIA EXTRAORDINARIA 2021/2022**

**Nombre:** Manuel Arroyo García

UO264184

## Contenido

Objetivo .....	2
Temática .....	2
HTML .....	2
CSS .....	3
XML.....	3
Computación ECMAScript en el cliente.....	4
Computación PHP en el servidor.....	5
Pruebas.....	6

## Objetivo

El proyecto de la convocatoria extraordinaria debe ser un proyecto web completo de una temática seleccionada por el estudiante (no se admiten ejercicios/módulos independientes) que contemple los módulos especificados en la guía docente de la asignatura.

El proyecto desarrollado y su temática deberán ser diferentes de los ejercicios abordados en las prácticas de la convocatoria ordinaria.

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. Se presentará en un único archivo en formato PDF.

## Temática

Resumen de la temática general del sitio web a desarrollar en el proyecto. Todos los módulos deben tener relación con la temática.

El objetivo del sitio web será proporcionar al cliente un sitio web donde pueda vender y administrar sus productos de hardware, principalmente ratones y teclados mecánicos.

La principal funcionalidad que ofrecerá esta página web será:

- Sistema de autenticación de usuarios y gestión de roles (Cliente, Admin).
- Apartado de introducción a la tienda y apartado de introducción a la tecnología que se ofrece.
- CRUD de productos.
- Sistema de compra donde el usuario podrá revisar su lista de productos y seleccionar el destino de envío.

## Apartado de soporte técnico donde el usuario podrá ponerse en contacto con el soporte técnico mediante email.HTML

Deben realizarse al menos 3 documentos HTML5 (ni generados ni compuestos a partir de incrustaciones de código) para el desarrollo del sitio web. Debe utilizarse el marcado semántico correcto, así como la estructuración correcta de los contenidos.

(Estos documentos son orientativos, tras el desarrollo es posible que sea necesario generar más documentos con el objetivo de crear un mejor producto)

**Index.html** Página inicial de la tienda a modo de presentación. Contendrá texto introductorio a la tienda junto a imágenes relevantes.

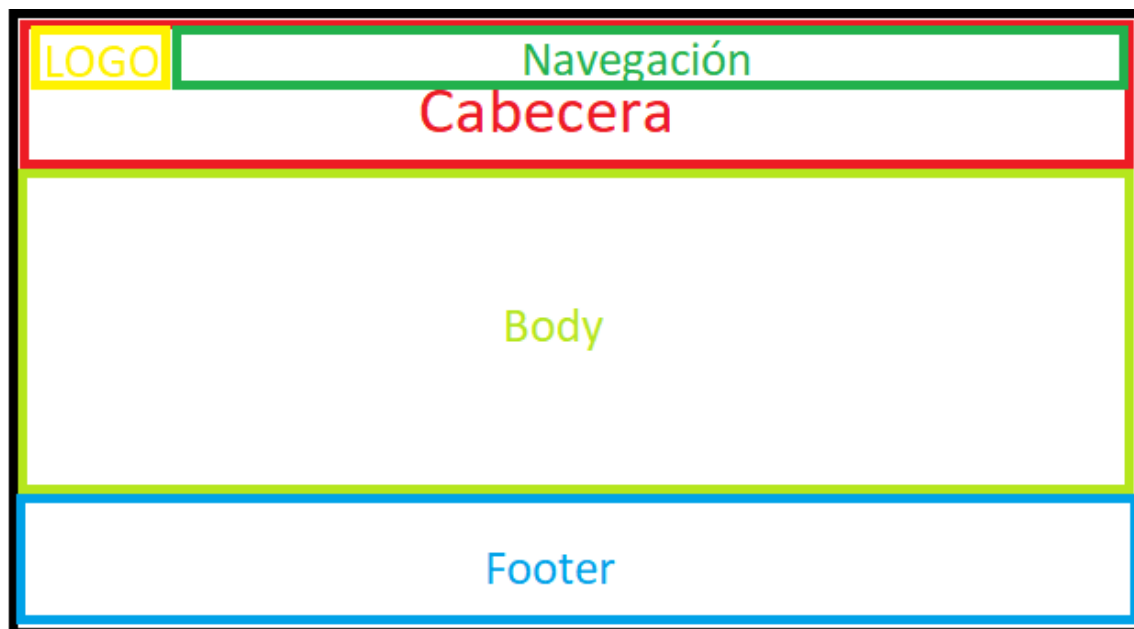
**Contact.html** Página con un formulario cuya acción será un servicio web que permite enviar correos electrónicos, el formulario contendrá el título, y contenido.

**Our\_products.html** Página información acerca de los diferentes tipos de teclados mecánicos se ofertan. La página contendrá imágenes junto con descripciones de los diferentes tipos de teclas que puede tener los teclados, además tendrá tabla comparativa entre los diferentes switches de las teclas.

## CSS

Deberá realizarse una (o varias) hojas de estilo para el sitio web propuesto. Debe realizarse una definición de selectores correcta y óptima, obviando el uso de ID y class salvo en casos debidamente justificados. Se debe garantizar la adaptabilidad del sitio web.

Las páginas HTML anteriormente mencionadas seguirán la siguiente estructura:



Para llevarlo a cabo, se utilizarán los siguientes archivos css:

- Estilo.css: Contendrá información relacionada con la paleta de colores a usar, estilo de la fuente, etc...
- Productos.xml Contendrá estilo especial para la lista de productos, posicionamiento de botones, disposición de los elementos de la lista, etc...
- Cart.css: Contendrá los estilos del mapa principalmente

## XML

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

- Deberá contener validadores DTD y Schema
- Deberá contener procesamiento de XML en ECMAScript utilizando jQuery

- Opcionalmente si se usan transformaciones XSLT no se pueden hacer con los navegadores (agentes de usuario) y será necesario documentar la herramienta con la que se ha realizado la transformación

El sitio Web podrá aceptar como mínimo los siguientes archivos:

- Pedido.xml Este archivo contendrá datos de los pedidos realizados por la tienda. Los nodos del pedido deberán tener los siguientes requisitos:
  - Número de pedido (por ejemplo 1)
  - Fecha en que se solicitó el pedido
  - Fecha de llegada del pedido
  - Lotes
    - Número de lotes (por ejemplo 1)
    - Fecha de creación de lote
    - Productos
      - Producto
        - Nombre (texto – Teclado 1)
        - Precio (Float – 100,3)
        - Imagen del producto (Enlace a archivo externo)
        - Descripción (texto – teclado con cherrys rojos)
        - Stock (True)
- Pedido.dtd: Este archivo será el validador DTD de Products.xml. Servirá para comprobar que se crea correctamente.
- Pedido.xsd: Este archivo será el validador Schema de Products.xml. Servirá para comprobar que se crea correctamente.

## Computación ECMAScript en el cliente

- Debe describirse la funcionalidad del proyecto para computación en el cliente que se desarrollará en ECMAScript.
- Deberá usarse obligatoriamente el paradigma de orientación a objetos. No se admitirá el paradigma procedimental ni otros paradigmas soportados por ECMAScript.
- No se pueden usar bibliotecas externas (a excepción de jQuery)
- Deberá usar jQuery
- Deberá usar JSON
- Deberá usar Ajax
- Deberá usar el API de HTML5 de Geolocalización
- Deberá usar otro API de HTML5
- Deberá consumir servicios Web. Es necesario especificar el sitio (o los sitios) web de donde se van a consumir los servicios web. Los servicios web utilizados no pueden ser los mismos que se usaron en las prácticas de evaluación continua.

El sitio Web contendrá como mínimo los siguientes archivos JS:

- Validador.js: Contendrá la clase validador, que tiene los métodos necesarios para validar el formulario de logueo o el formulario de registro.
- Cart.js: Contendrá la clase Cart con métodos necesarios para manejar el carrito, añadir producto, generar factura, eliminar item, etc...

- Producto.js: Contendrá una clase validador que consumirá servicios de php mediante el uso de Ajax y jQuery.

El sitio Web utilizará las siguientes APIs:

- API File de HTML5: Se usará para que el administrador pueda subir mediante ficheros XML nuevos productos a la plataforma.
- API de almacenamiento web de HTML5: Se usará para guardar la sesión del usuario.
- API de HTML5 de geolocalización: Se usará para ubicar al usuario en el mapa y así mostrarle información relevante a su zona.

El sitio Web consumirá los siguientes recursos:

- Leaflet: El sitio web consumirá mapas mediante la API de Leaflet para mostrar puntos de venta o para seleccionar el destino del envío. La información de puntos de venta será introducida por el administrador. Esta información deberá de ir en un fichero JSON.
- FormSubmit: El sitio Web se apoyará el formSubmit para enviar correos electrónicos al soporte de la tienda.

## Computación PHP en el servidor

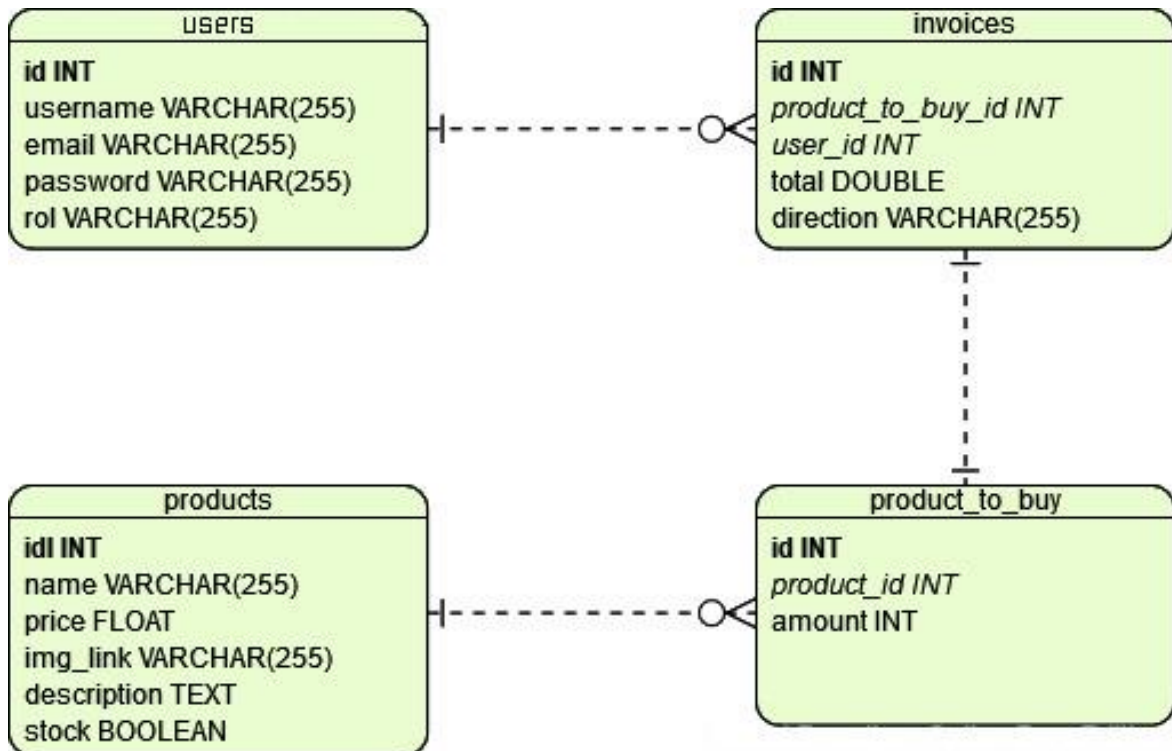
Debe describirse la funcionalidad del proyecto para computación en el servidor, que se desarrollará en PHP. Debe describirse la Base de Datos MySQL y cómo van a ser tratados estos para obtener informes, cálculos, etc. La Base de Datos debe tener un mínimo de 4 tablas relacionadas y normalizadas, y deberá entregarse el diagrama E-R de la misma.

- Deberá usarse el paradigma orientado a objetos obligatoriamente
- Deberá usarse el sistema de gestión de bases de datos en el servidor
- No se pueden usar bibliotecas externas

El usuario y password de la Base de Datos MySQL en XAMPP debe ser el mismo que en la convocatoria ordinaria.

El proyecto constará de las siguientes tablas en la base de datos:

- users: id, username, email, password, rol.
- invoices: id, user\_id, product\_to\_buy\_id, total, direction
- products\_to\_buy: id, product\_id, amount
- products: id, name, price, img\_link, description, stock



El proyecto dispondrá de varios archivos php encargados de registrar nuevos usuarios y acceder a ellos, lo mismo con la tabla de facturas (Invoice) y la de productos (Product):

- Session.php: Contendrá una clase que permitirá crear y devolver usuarios.
- Product.php. Contendrá una clase con métodos para implementar el CRUD de productos.
- Invoices.php: Contendrá una clase con métodos para crear facturas dada la información pertinente.

## Pruebas obligatorias a documentar en la Bitácora

Deberán realizarse las pruebas siguientes:

- Funcionamiento en máquina local mediante XAMPP
- Funcionamiento en la nube (AWS o Azure) con una cuenta de estudiante
- Validación del cumplimiento de los estándares HTML5 y CSS3, sin errores ni advertencias, mediante los validadores del W3C
- Pruebas de usabilidad del proyecto con al menos 12 personas indicando su nivel de destreza en la web, en 3 tandas de 4 personas. Se debe explicar en la Bitácora las tareas solicitadas a los usuarios y los tiempos que tardan en realizarlas. Entre las tandas se debe especificar en la Bitácora los cambios realizados para mejorar la usabilidad.
- Pruebas de adaptabilidad usando varios validadores.
- Pruebas de accesibilidad utilizando al menos 3 herramientas.

Todas las pruebas mencionadas anteriormente se realizarán y documentarán en el proyecto.