# Trabajo final de DWCS.

El trabajo final de la asignatura consiste en la creación de una aplicación web, realizada mediante el lenguaje PHP, para la gestión de **una ONG**.

Como el objetivo principal de la asignatura 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 normalizada y haber cumplido con las especificaciones que más adelante se detallan. A partir de ahí, el trabajo servirá para subir nota sobre el examen final del módulo, siempre y cuando se haya superado el mismo

La no realización del trabajo o la no superación del examen final supondrán una calificación negativa en el módulo.

### ONG Nombre\_ong

#### Requisitos funcionales.

A la aplicación web podrán acceder 4 tipos de usuarios:

* Usuarios anónimos (tipo 4). Un usuario anónimo es aquel usuario que no se haya identificado en la aplicación web y no posea una sesión activa.
* Usuarios registrados (tipo 3). Un usuario registrado es aquel que puede reservar productos que luego recogerá in situ en la tienda elegida (de un listado)
* Usuario gestor (tipo 2). Podrá dar de alta eventos y productos con los que se financiará la ONG, pero estos últimos no estarán visibles para su compra hasta que el usuario administrador los acepte o en el caso de los eventos haya una fecha o fechas definitivas para su realización.
* Usuarios administradores (tipo 1). Un usuario administrador tiene permisos de administración: aceptación de los productos; modificación de las características de los productos, tiendas y eventos; decisión final de las fechas de los eventos.

En el caso de usuarios tipo 1, 2, o 3 se deben almacenar el nombre completo, correo electrónico, teléfono, dirección completa y contraseña.

Los productos tendrán al menos los siguientes datos:

* Imagen pequeña para identificación y carrusel de imágenes. Todas las imágenes se guardarán en un sistema de carpetas de forma que cada producto constará de una carpeta diferenciada del resto.
* Características: nombre, descripción, unidades en stock, precio, categoría, fecha de caducidad, tiendas en las que se puede recoger, etc.
* Los productos guardarán una información sobre su estado, donde se definen al menos las siguientes categorías:

1. Propuesto. Es el estado por defecto que se asigna a un producto cuando se crea
2. Aceptado. El producto ya puede ser comercializado
3. Sin existencias.
4. Retirado. Por razones internas se decide retirar el producto y no se permite visualizarlo

* El administrador el único que lo puede pasar un producto a estado 2 y 4
* El producto pasa a estado sin existencias cuando el stock sea cero.

Un participante es una persona o grupo que actúa en un evento. Debe constar de nombre completo, correo electrónico, teléfono, NIF, dirección. Se debe indicar también si pertenece a un grupo.

Un grupo es un conjunto de participantes bajo un nombre común, CIF, domicilio fiscal y página web.

La tienda deberá guardar la siguiente información:

* Nombre, dirección completa con CP y ciudad, teléfono, correo electrónico y fax
* Se deberá mostrar en google maps su ubicación.

Un evento tiene aforo, puede tener o no participantes y se puede realizar varias veces en diferentes localizaciones y/o fechas. Por ejemplo una carrera solidaria puede tener dos fechas propuestas en dos ciudades distintas y no actúa nadie.

Operaciones:

* **Dar de alta un usuario**: Almacena los datos del usuario en la base de datos. Por defecto se crea con el perfil de tipo 3.
* **Identificación**: Operación que realiza un usuario para reservar un producto o una entrada a un evento; dar de alta un producto o evento y aprobar un evento o aceptar un producto en función de su perfil. Una vez identificado, la aplicación deberá guardar en la base de datos el momento del acceso.
* **Modificación del perfil del usuario**. Cualquier usuario puede modificar los datos asociados a él mismo. Se implementará con un trait.
* **Búsqueda de producto**: Operación que realizan los 4 tipos de usuarios. Se permitirá realizar una búsqueda por su categoría (A definir en la solución del trabajo, por ejemplo textil, alimentación,…) y sólo se mostrarán aquellos que estén en stockpara los usuarios tipo 4 y 3.
* **Reserva de un evento/producto**: Se descuentan unidades del aforo del evento/producto y se guarda en la base de datos las reservas que realiza un usuario concreto sobre ese evento/producto. Hay que registrar la fecha de la reserva, número de unidades, datos de los clientes, tienda en lo que recogerá, precio y para los usuarios de tipo 3 sólo se pueden mostrar aquellos eventos que todavía no se han realizado.
* **Búsqueda de un evento**: Se permitirá la búsqueda de un evento por calendario, entre un rango de fechas. Si es un usuario de tipo 4 y 3 sólo se mostrarán los eventos con fecha de realización.
* **Gestión de eventos/productos (backend)**: Operaciones realizadas por un usuario gestor y comprende de:
  + Dar de alta un producto
  + Dar de alta participantes en un evento, teniendo en cuenta que pueden estar integrados en un grupo, el cual definimos previamente.
  + Dar de alta un evento, en el que se especificará los participantes y un listado posible de fechas, junto con el lugar de celebración. Una fecha lleva asociada un único lugar de celebración
  + Modificar características de los eventos y productos.
  + Listado ordenado que relacione los productos/eventos reservados entre un rango de fechas y el usuario que hizo la operación
* **Administración de eventos/productos (backend)**: Operaciones realizadas por un usuario administrador y comprende de:
  + Modificar el estado de un producto
  + Modificar características de los eventos y productos.
  + Dar de altas y modificar los datos de las tiendas
  + Para un evento escoger una o varias fechas definitivas de entre las propuestas en las que se realizará
  + Listado ordenado que relacione los productos/eventos reservados entre un rango de fechas y el usuario que hizo la operación

#### Requisitos técnicos.

* Desarrollo en PHP bajo el paradigma de programación orientada a objetos (POO). Se deben crear las clases necesarias para que trabajen al menos con los usuarios, eventos, productos, tiendas y participantes. Dentro de este paradigma se pide:
  + Implementar una interfaz con tres métodos para dar de alta/baja/modificar productos y eventos. Ésta sólo la podrán implementar los usuarios de tipo 1 y 2.
  + Manejo de excepciones en PHP.
  + Empleo de transacciones
  + Implementar herencia. Por ejemplo para los usuarios y participantes.
  + Las clases deben estar en ficheros separados y definir namespaces
  + Implementar un trait para modificar el perfil del usuario y las características de un producto/evento
* Base de datos relacional mysqli con datos de ejemplo para su prueba, pero el acceso a los mismos es a través de PDO.
* Utilización de la interfaz de acceso a base de datos mediante objetos y utilización de sentencias preparadas.

#### Otras consideraciones.

* Debemos poblar la base de datos con al menos 10 tipos de eventos y productos, 3 tiendas, 15 participantes y al menos un usuario por tipo. En los productos las fotografías deben ser de similares características
* La parte cliente se debe hacer responsiva. Se recomienda usar bootstrap

#### Documentación.

La documentación debe ser de al menos 20 hojas (sin contar portadas) en donde se indicarán el dominio al que la hemos subido, los datos de acceso del usuario administrador, panel de control, junto con una descripción de la solución adoptada, su justificación y el modelo de entidad-relación de la base de datos. Además, habrá que proporcionar el código fuente en un fichero comprimido.

Por último, detallar las pruebas realizadas a las diferentes secciones de la aplicación indicando los resultados obtenidos.