

## **Trabajo final de DWCS- 2º DAW diurno**

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

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.

### **Requisitos funcionales.**

A la aplicación web podrán acceder 3 tipos de usuarios:

- Usuarios anónimos (tipo 3). Un usuario anónimo es aquel usuario que no se haya identificado en la aplicación web y no posea una sesión activa.
- Usuarios registrados (tipo 2). Un usuario registrado es aquel que puede ver los datos personales de los otros socios y dar de alta de alumnos y empresas en su zona de influencia. Siempre quedará identificado quien lo dio de alta, la fecha y será el responsable de esos datos. De forma que si alguien quiere ponerse en contacto con ese socio o empresa podrá dirigirse al usuario que los creó para poder solicitar más información.
- Usuarios administradores (tipo 1). Un usuario administrador tiene permisos de administración: cambio de roles de los usuarios gestores, etc.

En el caso de usuarios tipo 1 o 2 se deben almacenar el nombre completo, correo electrónico, contraseña, teléfono, dirección completa, código postal, país, género, institución a la que pertenece, web de la institución, número de teléfono de la institución, email de la institución, cargo dentro de esa institución y tipo de institución.

El objetivo de la web es poder disponer de una base de datos de socios en otros países que nos permita buscar empresas u otros socios en los que los alumnos podrán hacer las prácticas en el extranjero. Cada vez que un usuario introduce los datos de una empresa nueva recibe 20 puntos los cuales podrá canjear más adelante y será el responsable del mantenimiento de esos datos.

De una empresa o socio hay que guardar la siguiente información: tipo (socio, empresa, etc.), nombre completo, dirección, código postal, país, email, número de teléfono y las especialidades en las cuales se podría hacer prácticas (por ejemplo informática, electrónica). Para el socio se debe indicar su

género, datos de su institución, cargo dentro de esta y si puede gestionar la búsqueda de alojamiento para alumnos extranjeros. En la empresa además hay que indicar el socio que ha proporcionado el contacto y que será el responsable de mantener los datos actualizados, así como una pequeña descripción y a ser posible el VAT (número de identificación fiscal)

Cuando un usuario le solicita a otro la búsqueda del alojamiento y se lleva a cabo, el socio que realiza las gestiones del alojamiento recibe 15 puntos una vez que se registra la movilidad.

### Operaciones.

- **Dar de alta un socio:** Almacena los datos del usuario en la base de datos. Por defecto se crea con el perfil más bajo y recibe 10 puntos por registrarse. Hay que recoger la fecha de alta en el sistema. Se implementará con una interfaz de tipo CRUD
- **Dar de alta un alumno y/o empresa:** La podrán realizar sólo los usuarios registrados. Se recoge el nombre, dirección, email, nacionalidad, fecha de alta y especialidad. Para la empresa se deberá indicar además los datos del responsable (nombre completo, email, cargo, departamento, género, país y teléfono) y dar la opción de indicar su VAT (número de identificación fiscal), además de una pequeña descripción de la misma. Para el alumno su fecha de nacimiento y el socio responsable de su movilidad. Se implementará con una interfaz de tipo CRUD. El usuario que dé de alta una empresa recibe 20 puntos en su haber.
- **Solicitar dar de alta un determinada información:** Un usuario registrado cualquiera podrá enviar un email al administrador para dar de alta una determinada especialidad, o requerirle que modifique algo que él no puede hacer, etc.
- **Modificación del perfil de un socio, alumno y/o empresa.** Cualquier usuario registrado puede modificar los datos asociados a él mismo y de las empresas y/o alumnos que dependan de él. Se registrará la fecha de modificación y se implementará con una interfaz de tipo CRUD
- **Borrado de un socio o empresa.** Por motivos de mantener un historial, no se borrarán los datos, simplemente se recogerá la fecha de baja. Se implementará con una interfaz de tipo CRUD.
- **Búsqueda genérica de un socio/empresa por una especialidad y/o país:** Operación que realizar cualquiera. Mostrará sólo los datos básicos, que será para ambos el tipo, nombre, país y especialidad. Para las empresas se indicará también una descripción de la misma. Se implementará con una interfaz de tipo CRUD.
- **Búsqueda específica de un socio/empresa por una especialidad y/o país.** Igual que la anterior pero para usuarios registrados y mostrará toda la información de la que se dispone. Se implementará con una interfaz de tipo CRUD.
- **Registrar una movilidad:** Operación que relaciona el alumno con la empresa o socio en la cual va a realizar las prácticas. La debe consignar el socio responsable del alumno. Previamente se debe haber introducido los datos del alumno y tener dado de alta la empresa y/o socio. La aplicación nos presentará un menú que nos permitirá relacionar ambos y consumirá un saldo de 20 puntos. Se permitirá a un socio tener hasta un máximo de 30 puntos de déficit.

Si en el formulario se marca que el socio en el extranjero ha ayudado en la búsqueda del alojamiento éste recibirá 15 puntos por sus gestiones.

- **Histórico:** Ver los datos de las movilidades registradas entre unas fechas
- **Administración (backend):** Operaciones realizadas por un usuario administrador y comprende de:
  - Dar de alta una nueva especialidad
  - Dar de alta un tipo de institución
  - Listado ordenado que relacione socios nuevos entre un rango de fechas y el usuario que lo dio de alta, pudiendo mostrar el informe por país y/o especialidad
  - Listado de movilidades de alumnos entre un rango de fechas, pudiendo mostrar el informe por país y/o especialidad.
  - Enviar una petición a un socio de modificación o de completar los datos a su cargo a través de un email. Por ejemplo cuando los datos proporcionados no están en inglés.

### Requisitos técnicos.

- Desarrollo en PHP7 bajo el paradigma de programación orientada a objetos (POO). Se deben crear las clases necesarias para que trabajen al menos con los socios, alumnos y empresas. Dentro de este paradigma se pide:
  - Manejo de excepciones en PHP.
  - Empleo de transacciones
  - Implementar herencia, por ejemplo con los datos comunes de los socios, responsables, etc.
  - Las clases deben estar en ficheros separados y definir namespaces
  - Implementar elementos propios de PHP7 como pueden ser el operador coalesce, operador spaceship, sugerencias de tipo escalar, etc.
  - Utilizar Composer con PHPmailer para el envío de correos
  - Las password se debe guardar en la BBDD codificada
  - Utilización de cookies y sesiones
  - Empleo de polimorfismo en la búsqueda
- 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 para evitar inyección SQL

### Otras consideraciones.

- Debemos poblar la base de datos con al menos 15 tipos de socios, 5 alumnos, 10 empresas.
- La parte cliente se debe hacer responsiva. Se recomienda usar Bootstrap.
- Cada vez que se generan o consumen puntos hay que recoger la fecha en la que se llevó a cabo y la causa.

- Hay que subir la aplicación a un alojamiento web y poder conectarse a ella.
- Las puntuaciones se deben implementar en una tabla separada y tener un código

### 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. Se debe generar la documentación de las clases a través de phpDocumentor. Además, habrá que proporcionar el código fuente en un fichero comprimido.

### Información de interés:

Instalación de phpDocumentor con Composer: <https://docs.phpdoc.org/getting-started/installing.html#using-composer>

Ejecución de phpDocumentor: <https://docs.phpdoc.org/getting-started/your-first-set-of-documentation.html#writing-a-docblock>

Tags de phpDocumentor: [https://manual.phpdoc.org/HTMLSmartyConverter/HandS/phpDocumentor/tutorial\\_tags.pkg.html](https://manual.phpdoc.org/HTMLSmartyConverter/HandS/phpDocumentor/tutorial_tags.pkg.html)