

**Convocatoria:** Ordinaria

**Ciclo:** Desarrollo de Aplicaciones Web

**Módulo:** Desarrollo en entorno servidor

DATOS DEL ASPIRANTE			CALIFICACIÓN
APELLIDOS:			
NOMBRE:	D.N.I., N.I.E O Pasaporte	Fecha	

#### INSTRUCCIONES GENERALES PARA LA REALIZACIÓN DE LA PRUEBA

1. El examen práctico se entregará en un **archivo comprimido con el nombre del alumno/a en el apartado destinado para ello en el aula virtual del módulo, antes del fin de la hora prevista de entrega. No se aceptarán entregas fuera de plazo ni por otros medios que los aquí indicados.**
2. La profesora solo comprobará que se ha realizado una entrega en el aula virtual, **es responsabilidad del alumno/a comprobar y entregar los archivos correctos.**
3. La asistencia al examen tendrá como límite 10" de su hora estipulada, no permitiendo entrar en la reunión a posteriori.
4. Los micrófonos deberán estar silenciados. Solo la profesora podrá usarlo.
5. **Es obligatorio que** los alumnos conecten la cámara, o en su defecto utilicen la del móvil, y se enfoquen a sí mismos durante la realización del examen.
6. La reunión no será grabada.
7. Las dudas se plantearán y se contestarán por el chat privado de Webex.
8. **Durante el examen no se podrán realizar o contestar llamadas.**
9. Se pueden usar los recursos propios, o encontrados en internet que se consideren (**indicando en el código en forma de comentario el origen y una breve explicación de su funcionalidad**). No se considerarán válidos los de compañeros (puede dar lugar a sospecha de copia).
10. **Si la ejecución de un ejercicio produce algún tipo de error o aviso, su calificación será un 0.**
11. **La copia fraudulenta entre asistentes, así como la falsificación de la identidad dará lugar a la EXPLUSIÓN DEL EXAMEN Y/O SUSPENSO del alumno.**
12. Cualquier cosa que el alumno considere de interés explicar sobre el código deberá añadirse en forma de comentario. También deberán añadirse como comentarios dudas sobre la interpretación del enunciado y la solución adoptada.
13. No sólo se tendrá en cuenta la corrección del código, sino el uso correcto de lo aprendido a lo largo de la asignatura (código bien organizado, control de errores, etc.).

**EL NO CUMPLIMIENTO DE ESTAS NORMAS SUPONE EL SUSPENSO Y EXPULSIÓN DEL EXAMEN.**

**El test tiene un valor de 1,25 puntos, el examen práctico 8,75 puntos**

## ENUNCIADO DE LA PRUEBA

# Agenda de contactos

### Aspectos generales de la prueba:

- Crear una aplicación web que simule una agenda de contactos. Debe utilizarse **la versión 7.0 de PHP**.
- En todos los ficheros de código que se creen debe aparecer en la parte superior en forma de comentario el nombre del alumno/a.
- El título que aparecerá en las páginas web de la aplicación será **“Examen + nombre del alumno/a”**.
- No se valorará el uso de estilos CSS aunque sí que la aplicación tenga una presentación correcta.
- El código PHP debe incluir comentarios explicativos.
- Se permite, si se desea, el uso de Bootstrap para la mejora de la estética, pero no puntuará más.
- El uso de JavaScript **SOLO ESTÁ PERMITIDO** si se va a usar AJAX o jQuery para realizar consultas asíncronas.
- Junto con el enunciado se entrega una carpeta con imágenes y un archivo SQL que crea la base de datos y las tablas necesarias para la realización de los ejercicios.
- Para la conexión a la BD desde PHP se utilizará el usuario root contraseña “”.

### Ejercicio 1 (1,25 puntos):

- Crear un servicio SOAP (sin usar WSDL) que devuelva true o false si las credenciales (usuario y contraseña) que se le pasan por parámetro existen o no en la tabla “usuarios” de la BD.
- Las contraseñas se guardan en la base de datos cifradas con md5.
- Este ejercicio puntuará completo si en el ejercicio 2 se crea un cliente que lo utilice, y funciona correctamente.

### Ejercicio 2 (7,5 puntos):

En todas las páginas de la aplicación, excepto la del login, deberán aparecer SIEMPRE EN EL MISMO LUGAR, los siguientes elementos:

- El texto **“Hola”** + el nombre del usuario que se ha identificado.
- Un menú de enlaces con las siguientes opciones:
  - **Ver contactos:** al pulsarlo se accederá a la parte de listado de contactos.
  - **Cumpleaños:** al pulsar este enlace se accederá a la parte de cumpleaños.
  - **Desconectar:** el usuario se desconectará y se mostrará la página de login.

La aplicación constará de las siguientes partes:

- 1) **Login:** al iniciar la aplicación se pedirá mediante un formulario un nombre de usuario y una contraseña que se deberán cotejar con la base de datos pudiendo ocurrir los siguientes casos:
  - a. Si las credenciales no son correctas se mostrará el formulario de login apareciendo en él un mensaje resaltado que indique que los datos introducidos no son correctos, y el programa quedará a la espera de una nueva identificación (*NOTA: No es lo mismo admin que ADMIN o Admin*).
  - b. Si los datos son correctos el programa pasará a la parte de listado de contactos.

**NOTA:** Para la realización de esta parte, si se ha hecho el ejercicio 1 se debe crear un cliente que consuma el servicio creado.

2) **Listado de contactos:** Si se ha identificado un usuario correctamente, se accederá a la primera página de la aplicación (**en caso contrario, el programa debería redirigir al formulario de login**). En ella se mostrará un listado paginado de todos los contactos que hay en la BD. La información que se debe mostrar debe ser en este orden:

- Un encabezado: **“Contactos:”**
- La foto del contacto.** Si el contacto no tuviera foto en la BD se le asigna la imagen genérica **“avatar.png”**. *Se recomienda mostrar las imágenes con una altura de 40 píxeles y un ancho automático.*
- Nombre y apellidos** del contacto
- Teléfonos** del contacto (todos los que tenga). Puede ocurrir que no haya teléfono guardado de un contacto. En ese caso, la información del teléfono, aparecerá vacía.

3) **Cumpleaños:** Si se ha identificado un usuario correctamente, (**en caso contrario, el programa debería redirigir al formulario de login**) se mostrará un listado de los contactos que cumplen años **en la semana en la que se ejecute la aplicación (semanas diferentes, resultados diferentes)**, ordenados por fecha de más reciente a más antigua (se pueden usar funciones nativas de MYSQL en la consulta). El listado debe contener la siguiente información por este orden:

- Un encabezado: **“Cumplen años:”**
- La foto del contacto.** Si el contacto no tuviera foto en la BD se le asigna la imagen genérica **“avatar.png”**. *Se recomienda mostrar las imágenes con una altura de 40 píxeles y un ancho automático.*
- Nombre y apellidos del contacto.
- Fecha de nacimiento.
- La edad que tiene actualmente (calculada).

Si la fecha actual del servidor coincide con la fecha de nacimiento de un contacto, sus los datos se mostrarán de forma resaltada. En caso de que no haya contactos que cumplan años en la semana en curso se mostrará el mensaje **“no se han encontrado resultados”**.

#### Base de datos:

Para la conexión con la base de datos **“bdcontactos”** se usará el usuario **root** y la contraseña **“”**. El archivo SQL que os podéis descargar de la plataforma del curso crea la base de datos y las tablas con su contenido. Sus tablas son:

- usuarios:** contiene los datos de los usuarios que pueden acceder al sistema. Se compone de los campos:
  - usuario:** nombre del usuario.
  - Password:** contraseña cifrada con md5.

#	Nombre	Tipo
1	id_usuario	varchar(9)
2	password	varchar(50)

- contactos:** contiene un id de la tabla, nombre, apellidos, el nombre del fichero con el avatar del contacto, y un campo que indica la relación que tenemos con ese contacto.

#	Nombre	Tipo
<input type="checkbox"/> 1	id	int(11)
<input type="checkbox"/> 2	nombre	varchar(40)
<input type="checkbox"/> 3	apellidos	varchar(80)
<input type="checkbox"/> 4	foto	varchar(50)
<input type="checkbox"/> 5	relacion	varchar(15)

**fechas:** contiene un id de la tabla, un id del contacto y la fecha de nacimiento (campo de tipo DATE).

Nombre	Tipo
id 	int(11)
id_contacto	int(11)
fecha	date

- **teléfonos:** contiene un id de la tabla, un id del contacto, un campo teléfono. Un contacto puede tener más de un teléfono y un campo tipo que indica el tipo de teléfono que es (particular, trabajo, etc.)

Nombre	Tipo
id 	int(11)
id_contacto	int(11)
telefono	varchar(12)
tipo	varchar(15)

#### CRITERIOS DE CALIFICACIÓN Y VALORACIÓN

Los criterios de calificación se muestran en la siguiente tabla:

Funcionalidad	(0-3) (0%-50%)	(3-7) (50%-70%)	(8-10) (70%-100%)	Pts
Funcionalidad de la aplicación.	El sistema funciona y/o cumple los requisitos generales por debajo del 50%.	El sistema funciona y/o cumple los requisitos generales entre un 50% y un 70% de lo que debería.	El sistema se ejecuta correctamente y cumple todos los requisitos generales del enunciado.	3
Código	El código no es eficiente ni estructurado.	El código es parcialmente eficiente y estructurado.	El código es eficiente y estructurado.	0,5
Servicio SOAP y cliente para la identificación.	No crea ni consume el servicio o lo hace por debajo del 50%.	Crea el servicio y lo consume de forma incompleta o no es lo que se pedía.	Crea y consume el servicio de forma correcta.	1,25
Parte de login	No existe esta parte, o está por debajo del 50% de lo que se pedía.	La identificación es correcta de forma parcial.	La identificación es correcta.	0,4
Parte de listado de contactos	Esta parte está por debajo del 50%, o el listado no está paginado.	Muestra de forma parcial o incompleta la información, y/o no funciona bien la paginación.	Muestra el listado paginado completo y además usa consultas asíncronas para obtener los datos.	0,5
Parte de cumpleaños.	Esta parte está por debajo del 50% o no muestra los datos de la semana en que se ejecuta la aplicación.	Muestra los datos de forma incompleta o incorrecta.	Muestra toda la información que se pedía de forma correcta.	0,6

Preparación y acceso a BBDD	No se conecta con la BD o realiza menos del 50% de las operaciones que se pedían.	Realiza entre un 50% y un 70% de las operaciones que se piden y/o lo hace de forma parcial o incorrecta.	Realiza todas las operaciones necesarias con la BD de forma adecuada y eficiente, utilizando consultas preparadas, y optimizadas en SQL al trabajar con tablas	0,25
Sesiones	No usa sesiones o se producen errores o avisos al usarlas.	Usa sesiones pero no guarda la información adecuada o no la elimina cuando ya no es necesaria.	Usa sesiones correctamente	0,25
Diseño OO	No define clases o en su mayoría no están bien diseñadas aprovechando por debajo del 50% sus posibilidades.	Define clases aunque hay errores en el diseño de métodos y propiedades, siendo su funcionalidad básica, aprovechando sus posibilidades entre un 50 y un 70%.	Define clases y tanto su diseño y como la funcionalidad son correctos y eficientes.	0,5
Creación de vistas	No incluye el código de las vistas o el código aportado es muy pobre.	Se aportan vistas aunque hay errores en el diseño y funcionalidad.	Se aportan vistas con un diseño correcto y funcionales.	0,5
Diseño del código del controlador	Contempla menos del 20% de las peticiones posibles que pueden llegar a la aplicación. La estructura del código no es la adecuada.	Contempla menos del 80% de las peticiones posibles que pueden llegar a la aplicación. La estructura del código es la adecuada.	Contempla más del 80% de las peticiones posibles que pueden llegar a la aplicación. La estructura del código es la adecuada.	0,5
Creación de Modelos	No incluye el código de la parte de Modelo o el código es muy pobre.	Se incluye código de la parte de Modelo aunque hay errores en la funcionalidad y en el diseño o es insuficiente.	Se aporta código de la parte de Modelo con un diseño correcto y funcionales.	0,5