

Aclaración:

Se debe crear un archivo por cada entidad de PHP. Todos los métodos deben estar declarados dentro de clases. PDO es requerido para interactuar con la base de datos.

Se deben respetar los nombres de los archivos, de las clases, de los métodos y de los parámetros.

Parte BACKEND

Crear una aplicación Web que permita realizar un ABM de distintos usuarios y empleados.

Usuario.php. Crear, en `./backend/clases`, la clase **Usuario** con atributos públicos (id, nombre, correo, clave, id_perfil y perfil) y un método de instancia `ToJSON()`, que retornará los datos de la instancia nombre, correo y clave (en una cadena con formato **JSON**).

Agregar los siguientes métodos:

Método de instancia **GuardarEnArchivo()**, que agregará al usuario en `./backend/archivos/usuarios.json`. Retornará un **JSON** que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

Método de clase **TraerTodosJSON()**, que retornará un array de objetos de tipo Usuario, recuperado del archivo usuarios.json.

Método de instancia **Agregar()**: agrega, a partir de la instancia actual, un nuevo registro en la tabla **usuarios** (id,nombre, correo, clave e id_perfil), de la base de datos **usuarios_test**. Retorna **true**, si se pudo agregar, **false**, caso contrario.

Método de clase **TraerTodos()**: retorna un array de objetos de tipo Usuario, recuperados de la base de datos (con la descripción del perfil correspondiente).

Método de clase **TraerUno(\$params)**: retorna un objeto de tipo Usuario, de acuerdo al correo y clave que se reciben en el parámetro \$params.

Crear las siguientes páginas, en el directorio backend del proyecto:

AltaUsuarioJSON.php: Se recibe por POST el **correo**, la **clave** y el **nombre**. Invocar al método GuardarEnArchivo.

ListadoUsuariosJSON.php: (GET) Se mostrará el listado de todos los usuarios en formato JSON.

VerificarUsuarioJSON.php: (POST) Se recibe el parámetro **usuario_json** (correo y clave, en formato de cadena JSON) y se invoca al método TraerUno. Se retornará un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

AltaUsuario.php: Se recibe por POST el **correo**, la **clave**, el **nombre** y el **id_perfil**. Se invocará al método Agregar. Se retornará un **JSON** que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

ListadoUsuarios.php: (GET) Se mostrará el listado **completo** de los usuarios, exepto la clave (obtenidos de la base de datos) en una tabla (HTML con cabecera). Invocar al método TraerTodos.

Crear, en **./backend/clases**, la interface **IBM**. Esta interface poseerá los métodos:

Modificar: Modifica en la base de datos el registro coincidente con la instancia actual (comparar por id).

Retorna **true**, si se pudo modificar, **false**, caso contrario.

Eliminar (estático): elimina de la base de datos el registro coincidente con el id recibido cómo parámetro.

Retorna **true**, si se pudo eliminar, **false**, caso contrario.

Crear las siguientes páginas, en el directorio backend del proyecto:

ModificarUsuario.php: Se recibirán por POST los siguientes valores: **usuario_json** (id, nombre, correo, clave y id_perfil, en formato de cadena JSON), para modificar un usuario en la base de datos. Invocar al método Modificar.

Retornar un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

EliminarUsuario.php: Si recibe el parámetro **id** por POST, más el parámetro **accion** con valor "**borrar**", se deberá borrar el usuario (invocando al método Eliminar).

Retornar un JSON que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

Crear, en **./backend/clases**, la interface **ICRUD**. Esta interface poseerá los métodos:

TraerTodos (de clase): retorna un array de objetos de tipo Empleado, recuperados de la base de datos (con la descripción del perfil correspondiente y su foto).

Agregar (de instancia): agrega, a partir de la instancia actual, un nuevo registro en la tabla **empleados** (id,nombre, correo, clave, id_perfil, foto y sueldo), de la base de datos **usuarios_test**. Retorna **true**, si se pudo agregar, **false**, caso contrario.

Nota: La foto guardarla en **“./backend/empleados/fotos/”**, con el nombre formado por el nombre punto tipo punto hora, minutos y segundos del alta (Ejemplo: **juan.105905.jpg**).

Modificar (de instancia): Modifica en la base de datos el registro coincidente con la instancia actual (comparar por id). Retorna **true**, si se pudo modificar, **false**, caso contrario.

Nota: Si la foto es pasada, guardarla en **“./backend/empleados/fotos/”**, con el nombre formado por el nombre punto tipo punto hora, minutos y segundos del alta (Ejemplo: **juan.105905.jpg**). Caso contrario, sólo actualizar el campo de la base.

Eliminar (de clase): elimina de la base de datos el registro coincidente con el id recibido cómo parámetro.

Retorna **true**, si se pudo eliminar, **false**, caso contrario.

Empleado.php. Crear, en **./backend/clases**, la clase **Empleado** (hereda de Usuario) y posee atributos públicos (foto y sueldo). Implementa la interface **ICRUD**.

Crear las siguientes páginas, en el directorio backend del proyecto:

AltaEmpleado.php: Se recibirán por POST todos los valores: *nombre, correo, clave, id_perfil, sueldo* y *foto* para registrar un empleado en la base de datos.

Se retornará un **JSON** que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

ListadoEmpleados.php: (GET) Se mostrará el listado **completo** de los empleados (obtenidos de la base de datos) en una tabla (HTML con cabecera). Invocar al método TraerTodos.

Nota: *preparar la tabla (HTML) con una columna extra para que muestre la imagen de la foto (50px X 50px).*

ModificarEmpleado.php: Se recibirán por POST los siguientes valores: *empleado_json* (id, nombre, correo, clave, id_perfil, sueldo y pathFoto, en formato de cadena JSON) y *foto* (para modificar un empleado en la base de datos. Invocar al método Modificar.

Nota: *El valor del id, será el id del empleado 'original', mientras que el resto de los valores serán los del empleado modificado.*

Se retornará un **JSON** que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

EliminarEmpleado.php: Recibe el parámetro *id* por POST y se deberá borrar el empleado (invocando al método Eliminar).

Se retornará un **JSON** que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.