if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

    {

        // Transformamos el JSON de entrada de datos a un array asociativo

        $datos = json\_decode(file\_get\_contents('php://input'), true);

        $insert = "INSERT INTO mensajes(nombre, email, mensaje) VALUES (:nombre, :email, :mensaje)";

        $consulta = $conexion->prepare($insert);

        bindAllParams($consulta, $datos);

        $consulta->execute();

        $mensajeId = $dbConexion -> lastInsertId();

        if($mensajeId) {

            $datos['id'] = $mensajeId;

            salidaDatos(json\_encode($datos),

            array('Content-Type: application/json', 'HTTP/1.1 200 OK'));

        }

    }

if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

* Esta línea verifica si la solicitud HTTP recibida es de tipo POST. Esto es común cuando se reciben datos de un formulario o una solicitud Ajax. Solo si la solicitud es un POST se ejecutará el bloque de código dentro del if.

// Transformamos el JSON de entrada de datos a un array asociativo

$datos = json\_decode(file\_get\_contents('php://input'), true);

* file\_get\_contents('php://input') lee el contenido del cuerpo de la solicitud HTTP. Se asume que los datos enviados son en formato JSON.
* json\_decode(..., true) convierte el JSON a un array asociativo de PHP (el true indica que el resultado será un array y no un objeto).

$insert = "INSERT INTO mensajes(nombre, email, mensaje) VALUES (:nombre, :email, :mensaje)";

* Esta línea prepara una consulta SQL para insertar datos en una tabla llamada mensajes. Los valores :nombre, :email, y :mensaje son marcadores de posición que se rellenarán posteriormente con los valores reales.

$consulta = $conexion->prepare($insert);

* $conexion->prepare($insert) prepara la consulta SQL para su ejecución. prepare() es un método que prepara una consulta SQL para ser ejecutada de forma segura (en este caso, evitando inyecciones SQL).

bindAllParams($consulta, $datos);

* Aquí se llama a una función llamada bindAllParams, que no está definida en el fragmento de código que muestras, pero se puede asumir que esta función vincula los valores del array $datos a los parámetros de la consulta SQL (es decir, asigna valores a :nombre, :email, y :mensaje).

$consulta->execute();

* Esta línea ejecuta la consulta preparada, insertando los datos en la base de datos.

$mensajeId = $dbConexion->lastInsertId();

* Después de ejecutar la inserción, esta línea obtiene el ID del último registro insertado en la base de datos. Esto es útil si, por ejemplo, la tabla tiene una columna auto-incremental y queremos saber el ID asignado al nuevo mensaje.

if ($mensajeId) {

* Aquí se comprueba si se obtuvo un ID válido, es decir, si la inserción fue exitosa.

$datos['id'] = $mensajeId;

* Si la inserción fue exitosa, se agrega el ID del mensaje insertado al array $datos. Ahora, $datos tendrá una clave id que contiene el valor del ID del mensaje recién insertado.

salidaDatos(json\_encode($datos),

array('Content-Type: application/json', 'HTTP/1.1 200 OK'));

* json\_encode($datos) convierte el array $datos de nuevo a una cadena JSON.
* salidaDatos(...) parece ser una función que probablemente maneja la respuesta al cliente. En este caso, se le pasa el JSON codificado junto con los encabezados HTTP.
* Los encabezados indican que la respuesta será en formato JSON (Content-Type: application/json) y que la solicitud fue exitosa (HTTP/1.1 200 OK).

En resumen, este código maneja una solicitud POST que contiene un mensaje en formato JSON. Los datos del mensaje se insertan en la base de datos y, si la inserción es exitosa, el ID del mensaje recién insertado se agrega a la respuesta y se envía de vuelta al cliente como JSON.

Petición POST con un JSON con los identificadores del equipo y del alumno. En  
esta petición se tiene:  
◦ Comprobar que tanto el identificador del equipo como el del alumno existan en  
nuestro sistema. En caso de que alguno de ellos no exista, enviar la respuesta  
correspondiente.  
◦ Comprobar si el identificador del alumno ya está en otro equipo. En este caso  
se tiene que devolver una respuesta con el mensaje: “El alumno recibido ya  
está en otro equipo”.  
◦ Comprobar si el alumno tiene la edad mínima para estar en ese equipo de ese  
deporte.

◦ Comprobar que la edad del alumno es, como máximo, dos años más de la edad  
mínima. Si no se cumple esa condición no puede estar en ese equipo. Por  
ejemplo, si la edad mínima de un equipo es de 8 años, en ese equipo sólo  
puede haber alumnos que tengan 8, 9 o 10 años.  
◦ Comprobar si el equipo recibido tiene ya todos los jugadores (campo numero\_jugadores de tabla deportes). En caso de que  
no se puedan añadir más alumnos al equipo se debe devolver el siguiente  
mensaje: "Equipo con nombre “nombreEquipo” está completo y no admite  
más alumnos.", donde nombreEquipo es el nombre del equipo en el que  
queremos añadir al alumno.  
◦ En cualquier otro caso, dar de alta el alumno con el identificador recibido en el  
identificador de equipo recibido.