



# Funciones de la Pagina Web

Esta función verifica que los datos como correo o el nombre no existan

```
function verificarDuplicados(nombre, correo, callback) {  
const query = 'SELECT * FROM usuarios WHERE nombre = ? OR correo = ?';  
    const values = [nombre, correo];  
    pool.query(query, values, (err, results) => {  
        if (err) {  
            console.error('Error al verificar duplicados:', err);  
            return callback(err);  
        }  
        if (results.length > 0) {  
            return callback(null, true); // Hay duplicados  
        }  
        callback(null, false); // No hay duplicados  
    });  
}
```

# Create

```
function guardarUsuario(nombre, correo, contra, callback) {
  verificarDuplicados(nombre, correo, (err, duplicado) => {
    if (err) {
      return callback(err);
    }
    if (duplicado) {
      return callback(new Error('El nombre o el correo ya están registrados'));
    }
  });
  const query = 'INSERT INTO usuarios (nombre, correo, contra) VALUES (?, ?, ?)';
  const values = [nombre, correo, contra];
  pool.query(query, values, (err, results) => {
    if (err) {
      console.error('Error al guardar el usuario:', err);
      return callback(err);
    }
    console.log('Usuario guardado con éxito:', results);
    callback(null, results);
  });
}
```

# Read

## CONSULTAR TODOS LOS USUARIOS

```
function consultarUsuarios(callback) {  
  const query = 'SELECT * FROM usuarios';  
  pool.query(query, (err, results) => {  
    if (err) {  
      console.error('Error al consultar los usuarios:', err);  
      return callback(err);  
    }  
    console.log('Usuarios consultados con éxito:', results);  
    callback(null, results);  
  });  
}
```

## CONSULTAR POR ID (USUARIO)

```
function consultarUsuarioPorId(id, callback) {  
  const query = 'SELECT * FROM usuarios WHERE id = ?';  
  const values = [id];  
  pool.query(query, values, (err, results) => {  
    if (err) {  
      console.error('Error al consultar el usuario:', err);  
      return callback(err);  
    }  
    callback(null, results[0]); // Retornar el primer usuario encontrado  
  });  
}
```

# Update

## ACTUALIZAR

```
Function actualizarUsuario(id, nombre, correo, callback) {  
const query = 'UPDATE usuarios SET nombre = ?, correo = ? WHERE id = ?';  
    const values = [nombre, correo, id];  
    pool.query(query, values, (err, results) => {  
        if (err) {  
            console.error('Error al actualizar el usuario:', err);  
            return callback(err);  
        }  
        callback(null, results);  
    });  
}
```

# Delete

## ELIMINAR

```
function eliminarUsuario(id, callback) {  
  const query = 'DELETE FROM usuarios WHERE id = ?';  
  const values = [id];  
  pool.query(query, values, (err, results) => {  
    if (err) {  
      console.error('Error al eliminar el usuario:', err);  
      return callback(err);  
    }  
    if (results.affectedRows === 0) {  
      return callback(new Error('No se encontró un usuario con el ID  
proporcionado'));  
    }  
    console.log('Usuario eliminado con éxito:', results);  
    callback(null, results);  
  });  
}
```

## CONTRL DE INICIO DE SECCION

```
function iniciarSesion(correo, contra, callback) {
const query = 'SELECT * FROM usuarios WHERE correo = ? AND contra = ?';
    const values = [correo, contra];
    pool.query(query, values, (err, results) => {
        if (err) {
            console.error('Error al iniciar sesión:', err);
            return callback(err);
        }
        if (results.length === 0) {
            return callback(new Error('Correo o contraseña incorrectos'));
        }
        console.log('Inicio de sesión exitoso:', results[0]);
        callback(null, results[0]); // Retornar el usuario encontrado
    });
}
```

## GUARDAR RESPUESTAS DE UN TEST

```
function guardarRespuestasTest(usuario_id, tipo_test, respuestas, resultado,
    porcentaje, callback) {
const query = 'INSERT INTO respuestas_test (usuario_id, tipo_test, respuestas,
    resultado, porcentaje) VALUES (?, ?, ?, ?, ?)';
    const values = [usuario_id, tipo_test, JSON.stringify(respuestas), resultado,
        porcentaje];
    pool.query(query, values, callback);
}
```

## RESULTADOS DE TEST POR USUARIO

```
function obtenerResultadosPorUsuario(usuario_id, callback) {
const query = 'SELECT * FROM respuestas_test WHERE usuario_id = ? ORDER BY
    fecha DESC';
    pool.query(query, [usuario_id], callback);
}
```

## BUSCAR USUARIO POR CORREO

```
function buscarUsuarioPorCorreo(correo, callback) {  
  const query = 'SELECT * FROM usuarios WHERE correo =  
    ?';  
  pool.query(query, [correo], (err, results) => {  
    if (err) return callback(err);  
    if (results.length === 0) return callback(null, null);  
    callback(null, results[0]);  
  });  
}
```