

Actualización a API_PRODUCTOS/index.js:

```
1  ...      const express = require('express');
2           const morgan = require('morgan');
3           const path = require('path');
4           const app = express();
5  ...      const cors = require('cors');
6  ...      require('dotenv').config();
7  ...
8           //const PORT = 3000; //COMENTAR PARA CORRER BACK-END EN DOCKER

9  ...
10          // Middleware para logs y manejo de JSON
11          app.use(morgan('dev'));
12  ...      app.use(cors());
13  ...      app.use(express.urlencoded({ extended: true }));
14          app.use(express.json());
15          //
```

Código fue modificado a:

```
1  const express = require('express');
2  const morgan = require('morgan');
3  const path = require('path');
4  const app = express();
5  const cors = require('cors');
6  require('dotenv').config();
7
8
9  app.use(morgan('dev'));
10 app.use(cors());
11 app.use(express.urlencoded({ extended: true }));
12 app.use(express.json());
13 //eo
14 const database = require('./src/db');
15
16 const routes = require('./src/routes/index');
17 app.use(routes);
18
```

Actualización a API_CLIENTES\src\controllers\clientesController.js:

```
const getClienteById = (req, res) => {
  const clienteId = req.params.id;
  const query = 'SELECT * FROM clientes WHERE id_cliente = ?';

  database.query(query, [clienteId], (err, result) => {
    if (err) {
      console.error(err);
      res.status(500).send('Error al obtener el cliente por ID');
    } else {
      if (result.length === 0) {
        res.status(404).send('Cliente no encontrado');
      } else {
        res.json(result[0]);
      }
    }
  });
};
```

'If' statement should not be the only statement in 'else' block

Código fue actualizado a:

```
const getClienteById = (req, res) => {
  const clienteId = req.params.id;
  const query = 'SELECT * FROM clientes WHERE id_cliente = ?';

  database.query(query, [clienteId], (err, result) => {
    if (err) {
      console.error(err);
      return res.status(500).send('Error al obtener el cliente por ID');
    }
    if (result.length === 0) {
      return res.status(404).send('Cliente no encontrado');
    }
    res.json(result[0]);
  });
};
```

```

const updateCliente = (req, res) => {
  const clienteId = req.params.id;
  const { nombre_cliente, email, password } = req.body;

  const query =
    'UPDATE clientes SET nombre_cliente = ?, email = ?, password = ? WHERE id_cliente = ?'
  ;

  database.query(query, [nombre_cliente, email, password, clienteId], (err, result)
  ) => {
    if (err) {
      console.error(err);
      res.status(500).send('Error al actualizar el cliente');
    } else {
      if (result.affectedRows === 0) {

        res.status(404).send('Cliente no encontrado');
      } else {
        res.send('Cliente actualizado');
      }
    }
  });
};

```

'If' statement should not be the only statement in 'else' block

Código fue actualizado a:

```

const updateCliente = (req, res) => {
  const clienteId = req.params.id;
  const { nombre_cliente, email, password } = req.body;

  const query = 'UPDATE clientes SET nombre_cliente = ?, email = ?, password = ? WHERE id_cliente = ?';

  database.query(query, [nombre_cliente, email, password, clienteId], (err, result) => {
    if (err) {
      console.error(err);
      return res.status(500).send('Error al actualizar el cliente');
    }
    if (result.affectedRows === 0) {
      return res.status(404).send('Cliente no encontrado');
    }
    return res.send('Cliente actualizado');
  });
};

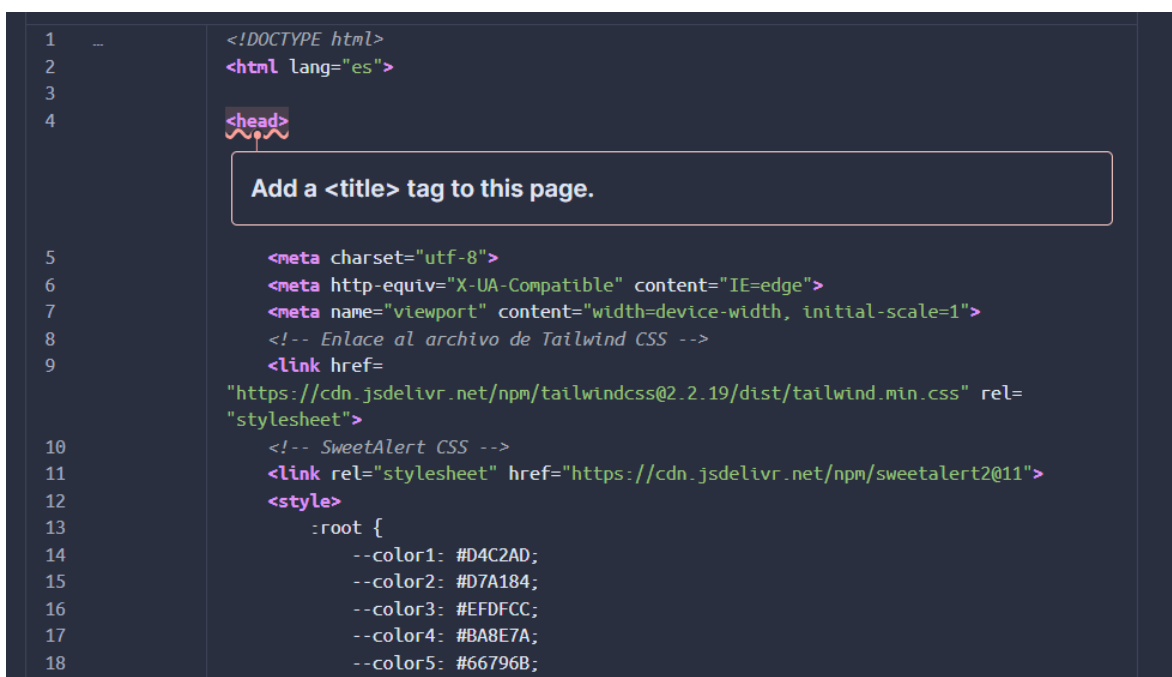
```


Actualización: API_CLIENTES/src/output.css



Código fue actualizado por completo para eliminar ese ese “issue”.

Actualización API_PRODUCTOS\public\crearpyme.html:



API_PRODUCTOS/public/crearpyme.html  [See all issues in this file](#)

```
1  --
2  <!DOCTYPE html>
3
4  <html lang="es">
5
6  <head>
7
8  <meta charset="utf-8">
9  <meta http-equiv="X-UA-Compatible" content="IE=edge">
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11 <!-- Enlace al archivo de Tailwind CSS -->
12 <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
```

Add a <title> tag to this page.

```
122         if (response.ok) {
123             const result = await response.json();
124
125             Swal.fire({
126                 icon: 'success',
```

Remove this useless assignment to variable "result".

```
122         if (response.ok) {
123             const result = await response.json();
124
125             Swal.fire({
126                 icon: 'success',
```

Remove the declaration of the unused 'result' variable.

Código fue modificado a :

```
API_PRODUCTOS > public > crearpyme.html > html
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5  <title>Crear Pyme</title>
6  <meta charset="utf-8">
7  <meta http-equiv="X-UA-Compatible" content="IE=edge">
8  <meta name="viewport" content="width=device-width, initial-scale=1">
9  <!-- Enlace al archivo de Tailwind CSS -->
10 <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
11 <!-- SweetAlert CSS -->
12 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@11">
13 <style>
14     :root {
15         --color1: #D4C2AD;
16         --color2: #D7A184;
17         --color3: #EFD9CC;
18         --color4: #BA8E7A;
19         --color5: #66796B;
20     }
```

```
Click to add a breakpoint
124
125
126
127
128
129
130
131
132

if (response.ok) {
  Swal.fire({
    icon: 'success',
    title: 'Éxito',
    text: 'Su pyme se ha registrado correctamente.',
  });
} else {
  throw new Error('Failed to fetch');
}
} catch (error) {
```

Actualización a API_PRODUCTOS\src\controllers\ProductosController.js:

```
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

const getProductosById = (req, res) => {
  const productId = req.params.id;
  const query = 'SELECT * FROM productos WHERE id_prod = ?';

  database.query(query, [productId], (err, result) => {
    if (err) {
      console.error(err);
      res.status(500).send('Error al obtener el producto por ID');
    } else {
      if (result.length === 0) {

        res.status(404).send('Producto no encontrado');
      } else {

        res.json(result[0]);
      }
    }
  });
}
```

'If' statement should not be the only statement in 'else' block

```
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77

const updateProducto = (req, res) => {
  const productId = req.params.id;
  const { nombre_producto, stock, precio, descripcion, etiqueta } = req.body;
  const query =
    'UPDATE productos SET nombre_producto = ?, stock = ?, precio = ?, descripcion = ?,
    etiqueta = ? WHERE id_prod = ?'
  ;

  database.query(query, [nombre_producto, stock, precio, descripcion, etiqueta,
    productId], (err, result) => {
    if (err) {
      console.error(err);
      res.status(500).send('Error al actualizar el producto');
    } else {
      if (result.affectedRows === 0) {

        res.status(404).send('Producto no encontrado');
      } else {

        res.send('Producto actualizado');
      }
    }
  });
}
```

'If' statement should not be the only statement in 'else' block

Código fue modificado a:

```
42  ✓ const getProductosById = (req, res) => {
43      const productId = req.params.id;
44      const query = 'SELECT * FROM productos WHERE id_prod = ?';
45
46  ✓  database.query(query, [productId], (err, result) => {
47  ✓      if (err) {
48          console.error(err);
49          return res.status(500).send('Error al obtener el producto por ID');
50      }
51  ✓      if (result.length === 0) {
52          return res.status(404).send('Producto no encontrado');
53      }
54
55      res.json(result[0]);
56  });
57
59  ✓ const updateProducto = (req, res) => {
60      const productId = req.params.id;
61      const { nombre_producto, stock, precio, descripcion, etiqueta } = req.body;
62      const query = 'UPDATE productos SET nombre_producto = ?, stock = ?, precio = ?,
63
64  ✓  database.query(query, [nombre_producto, stock, precio, descripcion, etiqueta, p
65  ✓      if (err) {
66          console.error(err);
67          return res.status(500).send('Error al actualizar el producto');
68      }
69  ✓      if (result.affectedRows === 0) {
70          return res.status(404).send('Producto no encontrado');
71      }
72
73      return res.send('Producto actualizado');
74
75      });
76
77
```

Actualización a API_PRODUCTOS/src/controllers/insumosController.js:

```
43     const getInsumoById = (req, res) => {
44         const insumoId = req.params.id;
45         const query = 'SELECT * FROM insumos WHERE id_insumo = ?';
46
47         database.query(query, [insumoId], (err, result) => {
48             if (err) {
49                 console.error(err);
50                 res.status(500).send('Error al obtener el insumo por ID');
51             } else {
52                 if (result.length === 0) {
53
54                     res.status(404).send('Insumo no encontrado');
55                 } else {
56                     res.json(result[0]);
57                 }
58             }
59         });
60     };

62     const updateInsumo = (req, res) => {
63         const insumoId = req.params.id;
64         const { nombre_insumo, stock, precio } = req.body;
65
66         const query =
67             'UPDATE insumos SET nombre_insumo = ?, stock = ?, precio = ? WHERE id_insumo = ?';
68
69         database.query(query, [nombre_insumo, stock, precio, insumoId], (err, result) => {
70             if (err) {
71                 console.error(err);
72                 res.status(500).send('Error al actualizar el Insumo');
73             } else {
74                 if (result.affectedRows === 0) {
75
76                     res.status(404).send('Insumo no encontrado');
77                 } else {
78                     res.send('Insumo actualizado');
79                 }
80             }
81         });
82     };
83 }
```

'If' statement should not be the only statement in 'else' block

'If' statement should not be the only statement in 'else' block

Código fue modificado a:

```
43 ∨ const getInsumoById = (req, res) => {
44   const insumoId = req.params.id;
45   const query = 'SELECT * FROM insumos WHERE id_insumo = ?';
46
47   database.query(query, [insumoId], (err, result) => {
48     if (err) {
49       console.error(err);
50       return res.status(500).send('Error al obtener el insumo por ID');
51     }
52     if (result.length === 0) {
53       return res.status(404).send('Insumo no encontrado');
54     }
55     return res.json(result[0]);
56   })
57 }
58 };
59
```

```
62 ∨ const updateInsumo = (req, res) => {
63   const insumoId = req.params.id;
64   const { nombre_insumo, stock, precio } = req.body;
65
66   const query = 'UPDATE insumos SET nombre_insumo = ?, stock = ?, precio = ? WHERE id_insumo = ?';
67
68   database.query(query, [nombre_insumo, stock, precio, insumoId], (err, result) => {
69     if (err) {
70       console.error(err);
71       return res.status(500).send('Error al actualizar el Insumo');
72     }
73     if (result.affectedRows === 0) {
74       return res.status(404).send('Insumo no encontrado');
75     }
76     return res.send('Insumo actualizado');
77   })
78 };
```

Actualización a API_PRODUCTOS/src/controllers/mensajesController.js:

```
41 const getMensajeById = (req, res) => {
42   const mensajeId = req.params.id;
43   const query = 'SELECT * FROM mensajes WHERE id_mensaje = ?';
44
45   database.query(query, [mensajeId], (err, result) => {
46     if (err) {
47       console.error(err);
48       res.status(500).send('Error al obtener el mensaje por ID');
49     } else {
50       if (result.length === 0) {
51
52         res.status(404).send('Mensaje no encontrado');
53       } else {
54         res.json(result[0]);
55       }
56     }
57   });
58 }
```

'If' statement should not be the only statement in 'else' block

```
59 const updateMensaje = (req, res) => {
60   const mensajeId = req.params.id;
61   const { nombre, email, mensaje } = req.body;
62
63   const query =
64     'UPDATE mensajes SET nombre = ?, email = ?, mensaje = ? WHERE id_mensaje = ?';
65
66   database.query(query, [nombre, email, mensaje, mensajeId], (err, result) => {
67     if (err) {
68       console.error(err);
69       res.status(500).send('Error al actualizar el mensaje');
70     } else {
71       if (result.affectedRows === 0) {
72
73         res.status(404).send('Mensaje no encontrado');
74       } else {
75         res.send('Mensaje actualizado');
76       }
77     }
78   });
79 }
```

'If' statement should not be the only statement in 'else' block

Código fue modificado a:

```
41  const getMensajeById = (req, res) => {  
42    const mensajeId = req.params.id;  
43    const query = 'SELECT * FROM mensajes WHERE id_mensaje = ?';  
44  
45    database.query(query, [mensajeId], (err, result) => {  
46      if (err) {  
47        console.error(err);  
48        return res.status(500).send('Error al obtener el mensaje por ID');  
49      }  
50      if (result.length === 0) {  
51        return res.status(404).send('Mensaje no encontrado');  
52      }  
53      return res.json(result[0]);  
54    })  
55  };
```

```
58  const updateMensaje = (req, res) => {  
59    const mensajeId = req.params.id;  
60    const { nombre, email, mensaje } = req.body;  
61  
62    const query = 'UPDATE mensajes SET nombre = ?, email = ?, mensaje = ? WHERE id_m';  
63  
64    database.query(query, [nombre, email, mensaje, mensajeId], (err, result) => {  
65      if (err) {  
66        console.error(err);  
67        return res.status(500).send('Error al actualizar el mensaje');  
68      }  
69      if (result.affectedRows === 0) {  
70        return res.status(404).send('Mensaje no encontrado');  
71      }  
72      return res.send('Mensaje actualizado');  
73    }  
74  }  
75  });
```

Actualización a API_PRODUCTOS/src/output.css:

```

433 [hidden] {
434     display: none;
435 }
436
437 *, ::before, ::after {
438     --tw-border-spacing-x: 0;
439     --tw-border-spacing-y: 0;
440     --tw-translate-x: 0;
441     --tw-translate-y: 0;
442     --tw-rotate: 0;
443     --tw-skew-x: 0;

```

Código fue modificado por completo.