

## 1. Agregar nuevas monedas

**Ruta:** POST /monedas

- Permite insertar una nueva conversión de moneda en la base de datos.
- Se espera un JSON con los siguientes campos:

json

CopiarEditar

```
{  
  "id": 1,  
  "origen": "USD",  
  "destino": "MXN",  
  "valor": 17.0  
}
```

- **Proceso:**
  - Usa `modeloMonedas.create()` para insertar la nueva moneda.
  - Si es exitoso, devuelve el objeto creado con estado HTTP 201 Created.
  - Si hay error (por ejemplo, datos inválidos), devuelve un mensaje de error con estado 500.

- **Código correspondiente:**

javascript

CopiarEditar

```
app.post('/monedas', async (req, res) => {  
  const { id, origen, destino, valor } = req.body;  
  
  try {  
    const nuevaEntrada = await modeloMonedas.create({  
id, origen, destino, valor });  
    res.status(201).json(nuevaEntrada);  
  } catch (error) {
```

```
        res.status(500).json({ error: 'No fue posible
agregar la moneda.' });
    }
});
```

## 2. Editar monedas

**Ruta:** PUT /monedas/:id

- Permite actualizar una conversión existente mediante su id.
- Se espera un JSON como:

json

CopiarEditar

```
{
  "origen": "USD",
  "destino": "EUR",
  "valor": 0.92
}
```

- **Proceso:**
  - Busca la moneda por su ID usando findByPk.
  - Si no se encuentra, responde con un 404.
  - Si se encuentra, actualiza los campos recibidos.
  - Devuelve la moneda modificada.

- **Código correspondiente:**

javascript

CopiarEditar

```
app.put('/monedas/:id', async (req, res) => {
  const { id } = req.params;
  const { origen, destino, valor } = req.body;

  try {
```

```

    const registroExistente = await
modeloMonedas.findByPk(id);

    if (!registroExistente) {
        return res.status(404).json({ error: 'Moneda no
encontrada.' });
    }

    registroExistente.origen = origen ??
registroExistente.origen;

    registroExistente.destino = destino ??
registroExistente.destino;

    registroExistente.valor = valor ??
registroExistente.valor;

    await registroExistente.save();

    res.json(registroExistente);
} catch (error) {
    res.status(500).json({ error: 'No se pudo modificar
la moneda.' });
}
});

```

El código cumple perfectamente con las dos funcionalidades solicitadas:

- **Agregar nuevas monedas** (POST /monedas).
- **Editar monedas existentes** (PUT /monedas/:id).

Puedes tomar este resumen y pegarlo directamente a tu documento Word. Si necesitas que te genere el código del modelo Sequelize (monedas.js) o el script SQL para la base de datos, dime y también te los preparo.

