**Implementación del Patrón Creacional Factory Method en Proyecto Node.js**

**1. Objetivo**

Implementar el **patrón creacional Factory Method** para crear instancias de *fetchers* que obtienen datos de diferentes APIs de criptomonedas. Esto permite agregar nuevas monedas o fuentes de datos sin modificar el código existente.

**2. Estructura de Archivos**

backend/

│

├─ index.js # Punto de entrada del servidor

├─ routes/

│ └─ cryptoRoutes.js # Definición de endpoints de criptomonedas

├─ fetchers/

│ ├─ CryptoFetcherFactory.js # Fábrica que devuelve fetchers según la moneda

│ └─ CoinMarketCapFetcher.js # Implementación concreta para CoinMarketCap

├─ package.json

└─ .env # Variables de entorno (API keys)

**3. Implementación Paso a Paso**

**3.1. CoinMarketCapFetcher.js**

Archivo que implementa la lógica para obtener datos de CoinMarketCap:

// CoinMarketCapFetcher.js

const fetch = require("node-fetch"); // Para v2 de node-fetch

require('dotenv').config();

class CoinMarketCapFetcher {

async getCryptoData(symbol) {

const response = await fetch(

`https://pro-api.coinmarketcap.com/v1/cryptocurrency/quotes/latest?symbol=${symbol}`,

{

headers: {

"X-CMC\_PRO\_API\_KEY": process.env.CMC\_API\_KEY

}

}

);

const data = await response.json();

return data;

}

}

module.exports = CoinMarketCapFetcher;

**3.2. CryptoFetcherFactory.js**

Fábrica que crea fetchers según la moneda o fuente de datos:

// CryptoFetcherFactory.js

const CoinMarketCapFetcher = require("./CoinMarketCapFetcher");

class CryptoFetcherFactory {

static getFetcher(source) {

switch (source) {

case "CMC":

return new CoinMarketCapFetcher();

// Se pueden agregar más fetchers aquí

default:

throw new Error("Fuente no soportada");

}

}

}

module.exports = CryptoFetcherFactory;

**3.3. cryptoRoutes.js**

Definición del endpoint que usa la fábrica:

// cryptoRoutes.js

const express = require("express");

const router = express.Router();

const CryptoFetcherFactory = require("../fetchers/CryptoFetcherFactory");

router.get("/crypto/:symbol", async (req, res) => {

try {

const fetcher = CryptoFetcherFactory.getFetcher("CMC");

const data = await fetcher.getCryptoData(req.params.symbol);

res.json(data);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

module.exports = router;

**3.4. index.js**

Inicialización del servidor Express:

// index.js

const express = require("express");

const app = express();

const cryptoRoutes = require("./routes/cryptoRoutes");

app.use("/api", cryptoRoutes);

app.listen(3000, () => {

console.log("Servidor corriendo en http://localhost:3000");

});

**4. Notas Importantes**

1. **Dependencias necesarias:**

npm install express node-fetch@2 dotenv

1. **.env**

CMC\_API\_KEY=TU\_API\_KEY\_DE\_COINMARKETCAP

1. **Node-fetch:**  
   Se usó la versión 2 para evitar problemas con require() en Node.js 20.