



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas



Ingeniería Telemática

Segunda Práctica de Aplicaciones Distribuidas

Bautista Uribe Juan Manuel

Aplicaciones distribuidas.

4TM3

CÓDIGO:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `Welcome`, `servidor.js`, and `package.json`.
- Editor:** The main editor tab is titled `servidor.js`. The code is as follows:

```
1 const express = require('express');
2 const axios = require('axios');
3 const cheerio = require('cheerio');
4
5 const app = express();
6 const PORT = 3000;
7
8 // URL OBJETIVO: Usaremos x-rates para buscar Libra Esterlina (GBP) a Peso Mexicano (MXN)
9 // Requerimiento 1 y 2: Investigar sitio y analizar estructura
10 const URL_COTIZACION = 'https://www.x-rates.com/calculator/?from=GBP&to=MXN&amount=1';
11
12 // Función auxiliar para hacer el Scraping
13 async function obtenerCotizacionOriginal() {
14     try {
15         // 1. Descargar el HTML de la página
16         const { data } = await axios.get(URL_COTIZACION);
17
18         // 2. Cargar el HTML en Cheerio para poder leerlo
19         const $ = cheerio.load(data);
20
21         // 3. Extraer el texto del precio
22         // NOTA: Inspeccioné la página y la clase que tiene el precio es "ccOutputRslt"
23         let precioTexto = $('.ccOutputRslt').text(); // Ejemplo: "22.50 MXN"
24
25         // Limpiamos el texto para que quede solo el número (quitamos " MXN")
26         let precioNumero = parseFloat(precioTexto.replace(' MXN', ''));
27
28         return precioNumero;
29     } catch (error) {
30         console.error("Error haciendo scraping:", error);
31         return null;
32     }
33 }
34
35 // Requerimiento 4: Servicio para recuperar valor de COMPRA
36 // Requerimiento 3: Alterar a la baja (Menos 10% para que el usuario compre)
37 app.get('/api/compra', async (req, res) => {
38     const precioReal = await obtenerCotizacionOriginal();
39
40     if (precioReal) {
41         // "Menos el 10% para que un usuario pueda comprar"
42         const precioCompra = precioReal * 0.90;
43
44         res.json({
45             moneda: "Libra Esterlina (GBP)",
46             accion: "Compra",
47             precio_mercado: precioReal,
48             precio_mejorado_cliente: precioCompra.toFixed(2),
49             mensaje: "¡Aprovecha! 10% más barato que el mercado."
50         });
51     } else {
52         res.status(500).send("Error obteniendo la cotización");
53     }
54 });
55
56 // Requerimiento 4: Servicio para recuperar valor de VENTA
57 // Requerimiento 3: Alterar al alza (Más 10% para que el usuario venda)
58 app.get('/api/venta', async (req, res) => {
59     const precioReal = await obtenerCotizacionOriginal();
60
61     if (precioReal) {
62         // "Un 10% más para que el usuario pueda vender"
63         const precioVenta = precioReal * 1.10;
64
65         res.json({
66             moneda: "Libra Esterlina (GBP)",
67             accion: "Venta",
68             precio_mercado: precioReal,
69             precio_mejorado_cliente: precioVenta.toFixed(2),
70             mensaje: "¡Te pagamos más! 10% por encima del mercado."
71         });
72 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `Welcome`, `servidor.js`, and `package.json`.
- Editor:** The main editor tab is titled `servidor.js`. The code is as follows:

```
1 const express = require('express');
2 const axios = require('axios');
3 const cheerio = require('cheerio');
4
5 const app = express();
6 const PORT = 3000;
7
8 // URL OBJETIVO: Usaremos x-rates para buscar Libra Esterlina (GBP) a Peso Mexicano (MXN)
9 // Requerimiento 1 y 2: Investigar sitio y analizar estructura
10 const URL_COTIZACION = 'https://www.x-rates.com/calculator/?from=GBP&to=MXN&amount=1';
11
12 // Función auxiliar para hacer el Scraping
13 async function obtenerCotizacionOriginal() {
14     try {
15         // 1. Descargar el HTML de la página
16         const { data } = await axios.get(URL_COTIZACION);
17
18         // 2. Cargar el HTML en Cheerio para poder leerlo
19         const $ = cheerio.load(data);
20
21         // 3. Extraer el texto del precio
22         // NOTA: Inspeccioné la página y la clase que tiene el precio es "ccOutputRslt"
23         let precioTexto = $('.ccOutputRslt').text(); // Ejemplo: "22.50 MXN"
24
25         // Limpiamos el texto para que quede solo el número (quitamos " MXN")
26         let precioNumero = parseFloat(precioTexto.replace(' MXN', ''));
27
28         return precioNumero;
29     } catch (error) {
30         console.error("Error haciendo scraping:", error);
31         return null;
32     }
33 }
34
35 // Requerimiento 4: Servicio para recuperar valor de COMPRA
36 // Requerimiento 3: Alterar a la baja (Menos 10% para que el usuario compre)
37 app.get('/api/compra', async (req, res) => {
38     const precioReal = await obtenerCotizacionOriginal();
39
40     if (precioReal) {
41         // "Menos el 10% para que un usuario pueda comprar"
42         const precioCompra = precioReal * 0.90;
43
44         res.json({
45             moneda: "Libra Esterlina (GBP)",
46             accion: "Compra",
47             precio_mercado: precioReal,
48             precio_mejorado_cliente: precioCompra.toFixed(2),
49             mensaje: "¡Aprovecha! 10% más barato que el mercado."
50         });
51     } else {
52         res.status(500).send("Error obteniendo la cotización");
53     }
54 });
55
56 // Requerimiento 4: Servicio para recuperar valor de VENTA
57 // Requerimiento 3: Alterar al alza (Más 10% para que el usuario venda)
58 app.get('/api/venta', async (req, res) => {
59     const precioReal = await obtenerCotizacionOriginal();
60
61     if (precioReal) {
62         // "Un 10% más para que el usuario pueda vender"
63         const precioVenta = precioReal * 1.10;
64
65         res.json({
66             moneda: "Libra Esterlina (GBP)",
67             accion: "Venta",
68             precio_mercado: precioReal,
69             precio_mejorado_cliente: precioVenta.toFixed(2),
70             mensaje: "¡Te pagamos más! 10% por encima del mercado."
71         });
72 }
```

The screenshot shows the Visual Studio Code interface with the 'servidor.js' file open in the editor. The code implements a simple API endpoint for a currency exchange application. It checks if the 'venta' parameter is present in the query string. If so, it calculates a 10% surcharge on the market price ('precioReal') and returns a JSON response with the result. If not, it returns an error message. The code also logs the server's port and the URLs for both the purchase and sale endpoints.

```
58 app.get('/api/venta', async (req, res) => {
59   const precioVenta = precioReal * 1.10;
60
61   res.json({
62     moneda: "Libra Esterlina (GBP)",
63     accion: "venta",
64     precio_mercado: precioReal,
65     precio_mejorado_cliente: precioVenta.toFixed(2),
66     mensaje: "¡Te pagamos más! 10% por encima del mercado."
67   });
68 } else {
69   res.status(500).send("Error obteniendo la cotización");
70 }
71 );
72
73 // Iniciar el servidor
74 app.listen(PORT, () => {
75   console.log(`Servidor de cotizaciones corriendo en http://localhost:${PORT}`);
76   console.log(`Prueba Compra: http://localhost:${PORT}/api/compra`);
77   console.log(`Prueba Venta: http://localhost:${PORT}/api/venta`);
78 });
79
80
81
82 }
```

EVIDENCIA:

Compra:

A screenshot of a web browser window displaying the JSON response from the '/api/compra' endpoint. The response object contains details about the purchase, including the currency ('Libra Esterlina (GBP)'), the action ('Compra'), the market price ('24.319936'), the improved price for the client ('21.89'), and a message encouraging the user to take advantage of the 10% discount.

```
{
  "moneda": "Libra Esterlina (GBP)",
  "accion": "Compra",
  "precio_mercado": 24.319936,
  "precio_mejorado_cliente": "21.89",
  "mensaje": "¡Aprovecha! 10% más barato que el mercado."
}
```

Venta:

A screenshot of a web browser window displaying the JSON response from the '/api/venta' endpoint. The response object contains details about the sale, including the currency ('Libra Esterlina (GBP)'), the action ('Venta'), the market price ('24.319936'), the improved price for the client ('26.75'), and a message stating that they pay 10% more than the market price.

```
{
  "moneda": "Libra Esterlina (GBP)",
  "accion": "Venta",
  "precio_mercado": 24.319936,
  "precio_mejorado_cliente": "26.75",
  "mensaje": "¡Te pagamos más! 10% por encima del mercado."
}
```