

UNIVERSIDAD ABIERTA INTERAMERICANA



FACULTAD DE TECNOLOGÍA INFORMÁTICA

CARRERA: Ingeniería en Sistemas Informáticos

PROFESOR: Ing. Sabato, Santiago.

GRUPO: Pauloni, Ariel;
Sandez, Agustín;
Zubini, Bruno.

MATERIA: Modelos Computacionales de Gestión Administrativa

Trabajo Práctico: Rest API Cotizaciones.

Fecha: 09/12/2019

Contenido del presente documento.

Funcionamiento del servicio.	3
Arquitectura del proceso.	6
Tabla de la Base de Datos.	6
Capturas del código.	7
<i>Página Welcome.aspx</i>	7
<i>ServicioCoizaciones.asmx</i>	8
<i>WebAPI. Controlador Cotizaciones</i>	9

Funcionamiento del servicio.

El Banco Central de la República Argentina cuenta con un servicio REST, el cual expone públicamente, que contiene un método, entre tantos otros, que devuelve un promedio de las cotizaciones del dólar desde el año 2000 hasta el día anterior a la fecha de consulta. Estos servicios se encuentran documentados en el siguiente link: <https://estadisticasbcra.com/api/documentacion> y, una vez pedido un TOKEN de conexión, se pueden realizar consultas a sus métodos, con un límite máximo de 100 consultas diarias a cualquiera de sus métodos.

El método que vamos a utilizar (más adelante se detallará qué uso le daremos), es el método <https://api.estadisticasbcra.com/usd>, cuya respuesta a la consulta API, devuelve en formato JSON con la siguiente estructura:

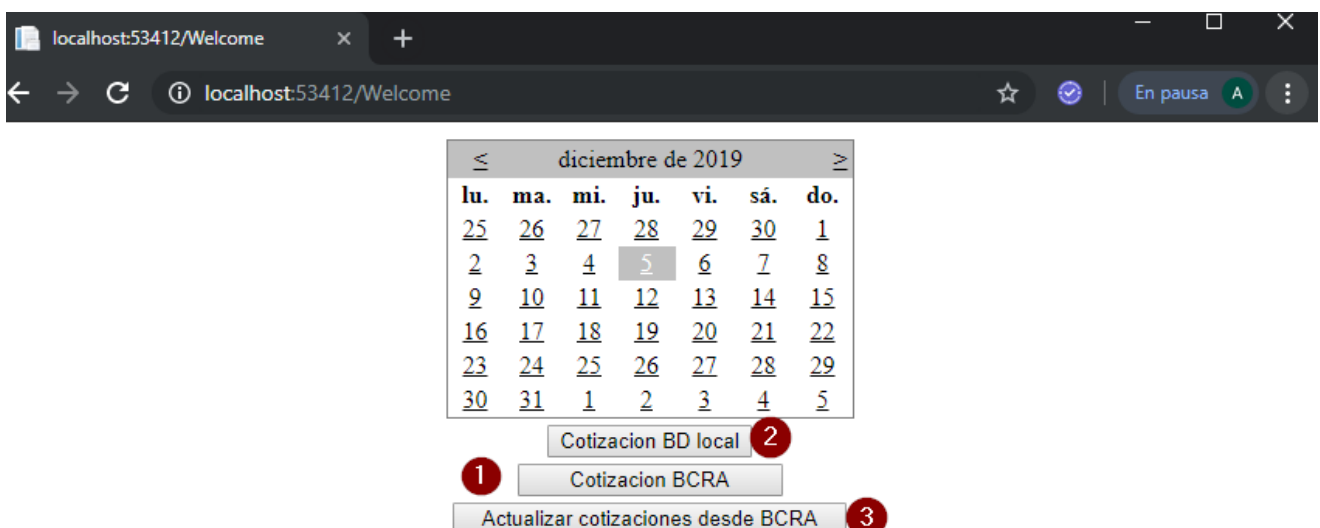
```
[
  {
    "d": fecha en formato MySQL,
    "v": valor del indicador para esa fecha.
  }
]
```

En el caso del método que consumimos, es un array de estos objetos.

El programa que planteamos hacer tiene, principalmente, las siguientes finalidades:

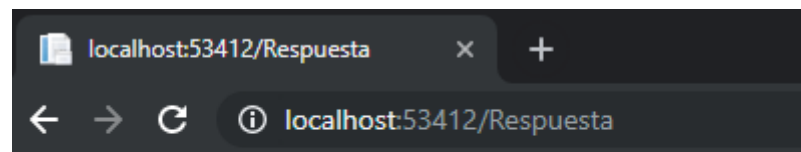
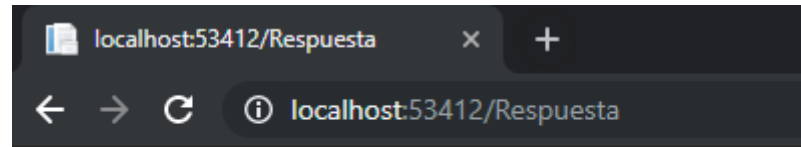
- 1) Mostrar cómo consumir una web API REST desde un servicio expuesto públicamente en la web.
- 2) Cómo construir un servicio REST propio para poder traer datos desde una base de datos propia.
- 3) Cómo consultar un servicio propio que actualice datos en la base de datos.

Para esto, la pantalla inicial contiene tres botones, cada uno de los cuales, termina satisfaciendo una de estas finalidades.



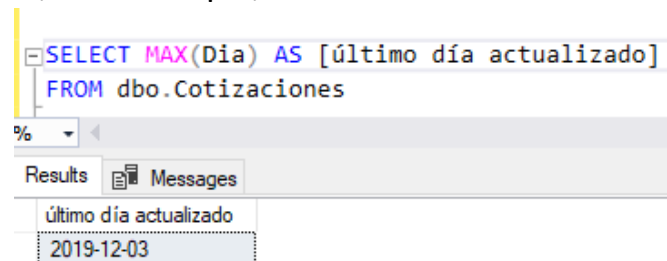
- 1) Desde esta funcionalidad, se consulta al servicio web del Banco Central de la República Argentina, que ya hemos explicado y, según la fecha seleccionada, nos devuelve la cotización promedio del dólar en esa fecha consultada. En caso de no haber cotización alguna (por ejemplo, si se consulta por un día no hábil), devuelve que no pudo obtenerla.

Por ejemplo, se consulta por la cotización del jueves 05 de diciembre del corriente año y el mismo día del año pasado.

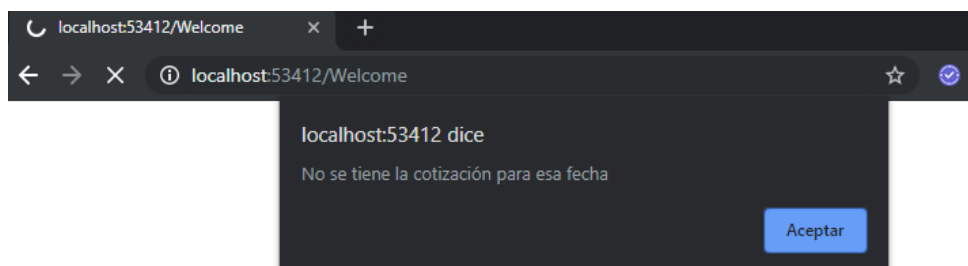


- 2) Desde este botón, se consulta a la base de datos local, la cual contiene la información previamente traída desde el servicio web del BCRA. El comportamiento es igual al caso anterior, la única diferencia es que puede resultar que no esté actualizada la base de datos local, por lo cual, haya fechas que no tengan resultados.

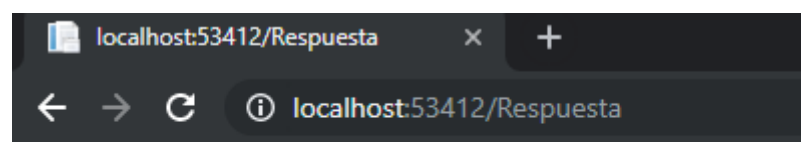
En este caso, consultando por las mismas fechas del ítem anterior, para la primera no se obtienen resultados, debido a que, la última fecha de actualización es del 03/12/19.



05/12/19:

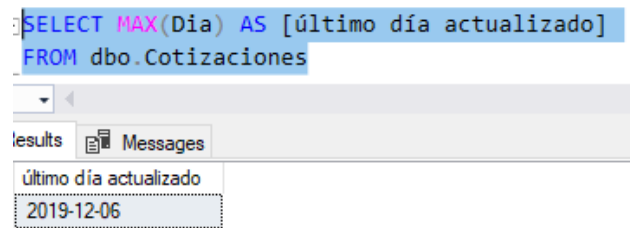


05/12/18:



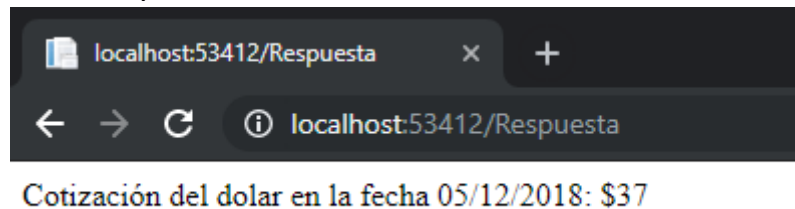
- 3) Este botón lo que hace es actualizar los datos locales, con los que trae desde el web service. Es decir, consulta el servicio del BCRA y, con esos datos, actualiza la base de datos local.

```
SELECT MAX(Dia) AS [último día actualizado]
FROM dbo.Cotizaciones
```



último día actualizado
2019-12-06

Se consulta nuevamente por el 05/12/19 desde el botón del servicio local:



Arquitectura del proceso.

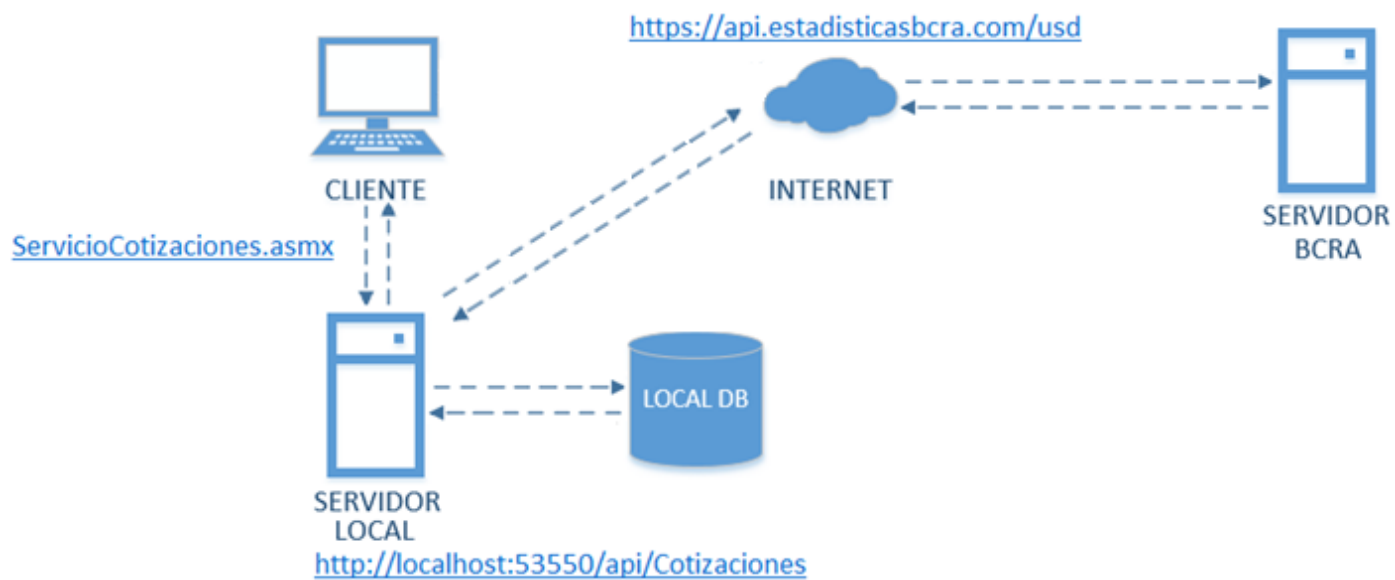


Tabla de la Base de Datos.

Cotizaciones	
	IdCotizacion
	Dia
	Valor

Capturas del código.

Página Welcome.aspx

```
11 2 referencias
12 public partial class Welcome : System.Web.UI.Page
13 {
14     0 referencias
15     protected void Page_Load(object sender, EventArgs e)
16     {
17         if (!IsPostBack)
18         {
19             calDia.SelectedDate = DateTime.Now;
20         }
21     }
22     0 referencias
23     protected void btnObtener_Click(object sender, EventArgs e)
24     {
25         if (calDia.SelectedDate != null)
26         {
27             if (calDia.SelectedDate > DateTime.Now)
28             {
29                 ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('' + " No puede obtener cotizaciones futuras, no somos adivinos" + '');", true);
30             }
31             else
32             {
33                 ServicioCotizaciones miServicio = new ServicioCotizaciones();
34                 decimal cotizacion = miServicio.CotizacionDolar(calDia.SelectedDate);
35                 if (cotizacion > 0)
36                 {
37                     Session["fecha"] = calDia.SelectedDate.ToString();
38                     Session["cotizacion"] = cotizacion.ToString("0.##");
39                     Response.Redirect("Respuesta.aspx");
40                 }
41             }
42             else
43             {
44                 ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('' + "No se tiene la cotización para esa fecha" + '');", true);
45             }
46         }
47         else
48         {
49             ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('' + "Seleccione una fecha por favor" + '');", true);
50         }
51     }
52     0 referencias
53     protected void btnCotizacionBCRA_Click(object sender, EventArgs e)
54     {
55         if (calDia.SelectedDate != null)
56         {
57             if (calDia.SelectedDate > DateTime.Now)
58             {
59                 ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('' + " No puede obtener cotizaciones futuras, no somos adivinos" + '');", true);
60             }
61             else
62             {
63                 List<CotizacionBCRA> cotizaciones = new List<CotizacionBCRA>();
64                 string text = System.IO.File.ReadAllText(Server.MapPath("AuthenticationBCRA_Service.txt"));
65                 HttpClient client = new HttpClient();
66                 client.DefaultRequestHeaders.Add("Authorization", text);
67                 string jsonResp = client.GetStringAsync("https://api.estadisticasbcra.com/usd").Result;
68                 cotizaciones = JsonConvert.DeserializeObject<List<CotizacionBCRA>>(jsonResp);
69                 var query = from a in cotizaciones where a.D == calDia.SelectedDate.ToString("yyyy-MM-dd").Substring(0, 10) select a;
70                 CotizacionBCRA cotizacion = query.FirstOrDefault();
71             }
72             if (cotizacion != null)
73             {
74                 Session["fecha"] = calDia.SelectedDate.ToString();
75                 Session["cotizacion"] = cotizacion.V.ToString("0.##");
76                 Response.Redirect("Respuesta.aspx");
77             }
78             else { ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('' + "No se tiene la cotización para esa fecha" + '');", true); }
79         }
80     }
81 }
82
83 0 referencias
84 protected void btnActualizarCotizaciones_Click(object sender, EventArgs e)
85 {
86     string text = System.IO.File.ReadAllText(Server.MapPath("AuthenticationBCRA_Service.txt"));
87     HttpClient client = new HttpClient();
88     client.DefaultRequestHeaders.Add("Authorization", text);
89     string jsonResp = client.GetStringAsync("https://api.estadisticasbcra.com/usd").Result;
90     if (!string.IsNullOrWhiteSpace(jsonResp))
91     {
92         System.IO.File.WriteAllText(Server.MapPath("CotizacionesBCRA.json"), jsonResp);
93         ServicioCotizaciones miServicio = new ServicioCotizaciones();
94         int resp = miServicio.ActualizarCotizaciones(Server.MapPath("CotizacionesBCRA.json"));
95     }
96 }
97
98 100
99
100
101 }
```

ServicioCotizaciones.asmx

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Services;
6 using WebAPI_Server;
7 using System.Net.Http;
8 using Newtonsoft.Json;
9 using System.Net;
10 using System.Data;
11 using System.Data.SqlClient;
12 using System.Configuration;
13
14
15 /// <summary>
16 /// Descripción breve de ServicioCotizaciones
17 /// </summary>
18 [WebService(Namespace = "http://tempuri.org/")]
19 [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
20 // Para permitir que se llame a este servicio web desde un script, usando ASP.NET AJAX, quite la marca de comentario de la línea siguiente.
21 // [System.Web.Script.Services.ScriptService]
22 5 referencias
23 public class ServicioCotizaciones : System.Web.Services.WebService
24 {
25     private SqlConnection myConnection = new SqlConnection();
26     private readonly string ConnectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
27
28     2 referencias
29     public ServicioCotizaciones()
30     {
31
32     }
33
34     [WebMethod]
35     1 referencia
36     public decimal CotizacionDolar(DateTime fecha)
37     {
38         decimal ret = 0;
39         Cotizacion cotizacion = new Cotizacion();
40         string queryString = "http://localhost:53550/api/Cotizaciones?fecha=" + fecha.ToString("yyyy-MM-dd");
41
42         HttpClient client = new HttpClient();
43         var response = client.GetAsync(queryString);
44
45         if ((response.Result != null) && (response.Result.IsSuccessStatusCode))
46         {
47             var jsonResp = response.Result.Content.ReadAsStringAsync();
48             cotizacion = JsonConvert.DeserializeObject<Cotizacion>(jsonResp.Result);
49             if (cotizacion != null) { ret = cotizacion.Valor; }
50         }
51         return ret;
52     }
53
54     [WebMethod]
55     1 referencia
56     public int ActualizarCotizaciones(string filePath)
57     {
58         int ret = 0;
59         myConnection.ConnectionString = ConnectionString;
60         myConnection.Open();
61
62         SqlCommand sqlCmd = new SqlCommand("dbo.pr_ActualizarCotizaciones ", myConnection);
63         sqlCmd.Parameters.Add("@path", SqlDbType.VarChar).Value = filePath;
64         sqlCmd.CommandType = CommandType.StoredProcedure;
65         sqlCmd.ExecuteNonQuery();
66
67         if (myConnection != null && myConnection.State == ConnectionState.Open)
68             myConnection.Close();
69
70         return ret;
71     }
72 }
```



```
13 namespace WebAPI_Server.Controllers
14 {
15     0 referencias
16     public class CotizacionesController : ApiController
17     {
18         private DolarEntities db = new DolarEntities();
19
20         // GET: api/Cotizaciones
21         0 referencias
22         public IQueryable<Cotizaciones> GetCotizaciones()
23         {
24             return db.Cotizaciones;
25         }
26
27         // GET: api/Cotizaciones/5
28         [ResponseType(typeof(Cotizaciones))]
29         0 referencias
30         public IHttpActionResult GetCotizaciones(int id)
31         {
32             Cotizaciones cotizaciones = db.Cotizaciones.Find(id);
33             if (cotizaciones == null)
34             {
35                 return NotFound();
36             }
37
38             return Ok(cotizaciones);
39         }
40
41         // GET: api/Cotizaciones/5
42         [ResponseType(typeof(Cotizaciones))]
43         0 referencias
44         public IHttpActionResult GetCotizaciones(DateTime fecha)
45         {
46             var query = from a in db.Cotizaciones where a.Dia == fecha select a;
47             Cotizaciones cotizaciones = query.FirstOrDefault();
48             if (cotizaciones == null)
49             {
50                 return NotFound();
51             }
52
53             return Ok(cotizaciones);
54         }
55     }
56 }
```