Ejercicio - Entrevista - Dirmod v1.3

Fecha: 2020/02/09

Version: v1.3

**1** - Implementar una Web Api RESTful que retorne las cotizaciones de, al menos, 3 monedas.

* La misma debe estar programada en **C#** *.Net Core*, **Javascript**, **Python** o **Java** según tu postulación.La alineación del ejercicio y la postulación es Requerida, caso contrario se descartará la postulación.
* El proyecto debe ser subido a **GitHub** listo para descargar y correr.
* Cualquier aclaratoria debe estar en el archivo **README.md** del repositorio.
* <https://searchmicroservices.techtarget.com/definition/RESTful-API>
* <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-2.2&tabs=visual-studio>  
  <https://medium.freecodecamp.org/building-a-simple-node-js-api-in-under-30-minutes-a07ea9e390d2>

**REQUEST**

Http verb: GET  
URL: */cotizacion/dolar*  
**RESPONSE**  
*{  
 “moneda”: “dolar”,*

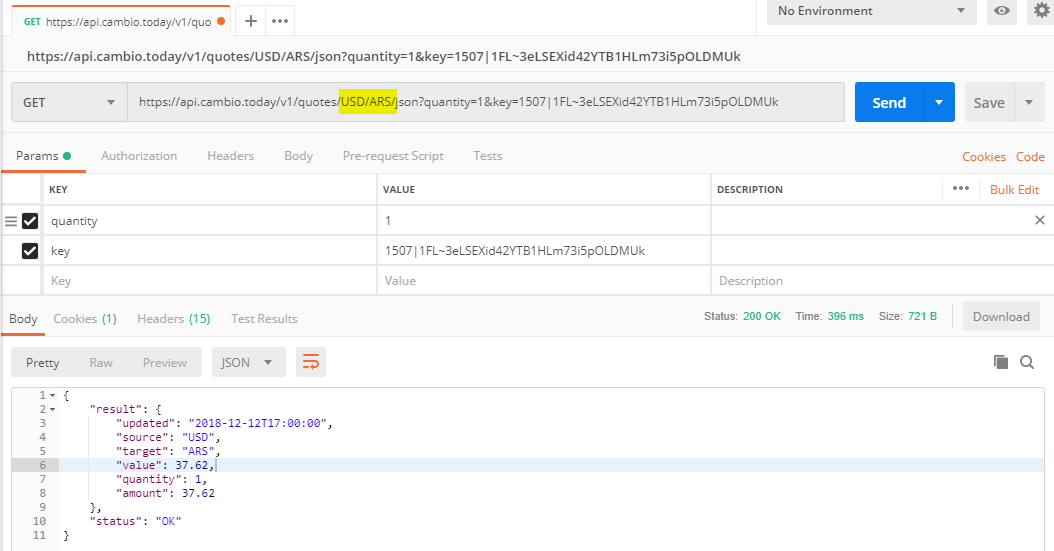
*“precio”: “38.2”   
}*

**1.B** Se deben implementar con el mismo formato de respuesta:   
“/cotizacion/euro” y   
“/cotizacion/real”.

**1.C** Obtener Cotizacion de el BCRA o api.cambio.

* Opcion A: <https://cambio.today/api/quotes/full>
* Opcion B: <http://estadisticasbcra.com/api/documentacion>
* Opcion C: Cualquier otra que tenga la misma función.

Ejemplo



Se recomienda y se puntuará positivamente el uso de:

**2** - Strategy Pattern

<https://es.wikipedia.org/wiki/Strategy_(patr%C3%B3n_de_dise%C3%B1o)#C#>

**3** - Codigo “self-explanatory” o ”self-documenting”  
<https://en.wikipedia.org/wiki/Self-documenting_code>

**4** - Aplicación del Principio SOLID  
<https://en.wikipedia.org/wiki/SOLID>

**5** - Implementación de FrontEnd para visualizar las 3 cotizaciones, refrescandose automáticamente cada **5** segundos.  
  
**6** - Uso de **ReactJs** y tecnologías como ***Redux*** y *WebPack*.

**7** - Implementación del WebSite en cualquier Nube: AWS, GCP, Azure, heroku o simil.