

Desafío Nubimetrics para el área de BackEnd SSR

1 - Introducción

En el área de backend de [Nubimetrics](#) tenemos múltiples desafíos. Los principales crear APIs para que distintas áreas puedan consumir esa información.

Para poder cubrir las responsabilidades de un perfil junior pensamos que deberías poder tener un entendimiento básico sobre el uso de patrones de diseños, testing y buenas prácticas de desarrollo.

Te queremos proponer un trabajo sobre la creación y consumo de diferentes endpoints para así poder entender tu nivel de conocimiento de .Net como así también tu compromiso en el desarrollo de una solución con un **código claro, limpio y ordenado**. Lo que es una de nuestras prioridades. Así también nos gustaría encontrar en tu solución una correcta **separación de responsabilidades** a la hora de proponer una solución.

Forma de Entrega:

- Link a [Github](#) con el código fuente.
- Script para generar la base de datos y luego popular en el caso que sea necesario.
- Un [documento de Google Drive](#) con las instrucciones para poder ejecutar la aplicación y servicios.
- Todos los desafíos deben estar en el mismo repositorio.

Framework permitidos: .Net 4.7 o .Net Core 2.2 en adelante

Debido a que estamos tratando de comprender tus conocimientos en .Net te pedimos por favor que **NO** uses el [software development kit \(SDK\)](#) para .NET de MercadoLibre: [Link al repositorio del SDK](#)

Bonus:

- Crear [unit test](#)
- Implementación de patrones de diseño y explicación, en la documentación, de por qué es relevante ese patrón para la solución propuesta.

2 - Desafíos

Desafío 1: Crear endpoint “países”

Crear una aplicación del tipo REST Web Api y exponer el endpoint que llamaremos “Países”

Este endpoint deberá ser llamado los parámetros del país. Por Ejemplo:

Para

- <http://localhost:8080/MyRestfulApp/Paises/PAIS>
- El País (PAIS) debería ser: "AR", "BR", "CO". La totalidad de los sites se pueden encontrar en la siguiente documentación de mercadolibre: [API classified Locations de Mercado Libre](#)

Consideraciones:

- Para los parámetros "BR" y "CO" hacer que el endpoint de una respuesta que sea **"error 401 unauthorized de http"**
- Para el parámetro "AR" el endpoint deberá consumir la información del país desde el servicio externo: [API classified Locations de Mercado Libre - Countries: AR](#)

Desafío 2: Crear endpoint "búsqueda":

Crear una aplicación del tipo REST Web Api exponer el endpoint que llamaremos "búsqueda"

El endpoint deberá ser llamado por un parámetro de tipo string. Por Ejemplo

Para:

- <http://localhost:8080/MyRestfulApp/busqueda/TERMINO>
- El término (TERMINO), por ejemplo Iphone, deberá ser lo que se debe buscar. El endpoint deberá consumir la información del siguiente servicio externo: [Resultado de la api search para el término 'Iphone' en Mercado Libre](#)

Consideraciones:

- Este endpoint deberá devolver el objeto como lo devuelve el servicio externo.
- En el array "results" solo incluir los Fields:
 - id
 - site_id
 - title
 - price
 - seller.id
 - permalink

Desafío 3: Crear endpoint "usuarios":

Crear una aplicación del tipo REST Web Api exponer el siguiente endpoint:

Crear un endpoint "usuarios" que devuelva todos los usuarios de una tabla "User" con los campos, id, nombre, apellido, email, password. Para ello crear la base de datos, la tabla y popularla con un script.

Crear los endpoints necesarios en la misma api rest para el ABM de usuarios.

Desafío 4: Consumir datos de endpoints

Al ser ejecutado el proyecto se debe poder consumir datos de los siguientes endpoints:

- Api Currencies de Mercado Libre: [Link a la API](#)
- Api Currency Conversion de Mercado Libre: [Link a la API](#)

El objetivo es que se logre almacenar en disco un json con la estructura que devuelve el endpoint [Api Currencies de Mercado Libre](#). Adicionalmente se debe incluir una nueva property "todolar" con el resultado del endpoint [Api Currency Conversion de Mercado Libre](#). Tené en cuenta que: El endpoint [Currency Conversion](#) toma como parámetro en "from" el id de moneda correspondiente a un país. Id que es devuelto por el endpoint [Currencies](#) - Para más información podés consultar la documentación de [Mercado Libre](#) sobre [Ubicación y Monedas](#)

Adicionalmente la misma aplicación tiene que almacenar en disco un archivo csv con cada uno de los resultados obtenidos de "currency_conversions", es decir debe almacenar sólo los resultados obtenidos de la property "ratio" (Ej: 0.0147275,0.013651,0.727565).