

Ejercicio Práctico 1 (C#)

Implementar una web API RESTful que:

1. Esté programada en C# (.NET)
2. Permita obtener las últimas cotizaciones de, por lo menos, cinco cryptomonedas.
 - Deben estar incluidas: Bitcoin (BTC), Ethereum (ETH), Binance Coin (BNB), Tether (USDT), Cardano (ADA)
 - El sitio deberá actualizar la información de la pantalla cada 5 segundos. (No tiene que hacer Postback)
3. Seleccionar una cryptomoneda, ingresar una cantidad y mostrar el monto final de la conversión a las cryptomonedas restantes.

El proyecto deberá:

- Estar subido a GitHub.
- Tener la API Key como un campo configurable (no hard-codeado en los consumos de API).
- Incluir cualquier comentario en el Readme.md del repositorio.

API a utilizar:

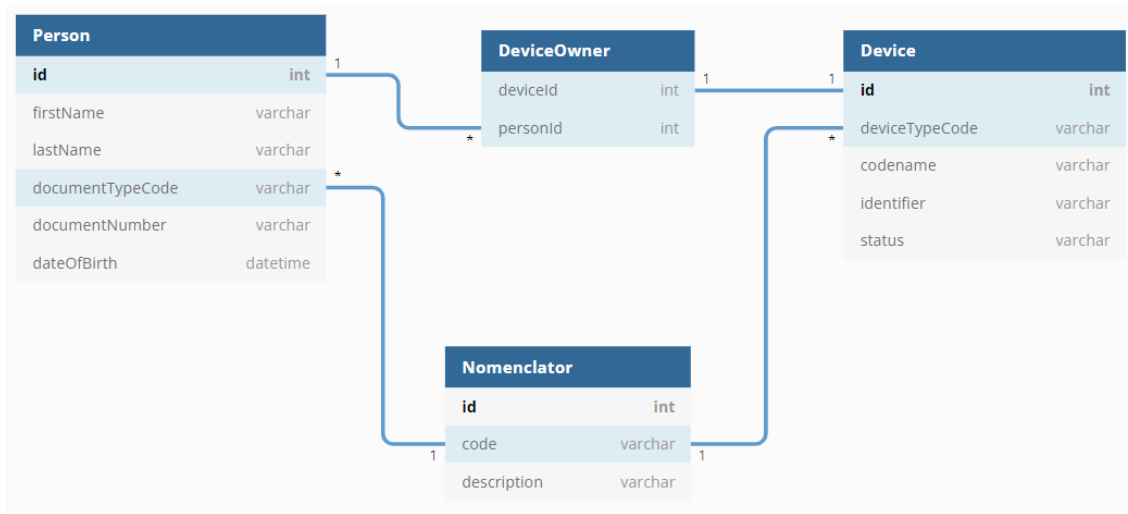
<https://coinmarketcap.com/api/documentation/v1/>

Se valorará positivamente:

- Uso de Angular.
- Manejo de errores.
- Utilizar Patrones de diseño.

Ejercicio práctico 2 (SQL)

Dado el siguiente esquema de tablas y ejemplos de datos:



Person					
id	firstName	lastName	documentTypeCode	documentNumner	dateOfBirth
1	Juan	Perez	DOCARG1	24847388	13/7/1975
2	Matias	Gonzalez	DOCARG2	94003984	14/2/1980
3	Alejandra	Gomez	DOCARG1	25154667	27/3/1977

DeviceOwner	
deviceId	personId
2	1
1	3
4	1

Nomenclator		
id	code	description
1	DEVMOB	Móvil personal
2	DEVMOBW	Móvil trabajo
3	DOCARG1	DNI
4	DOCARG2	DNI Extranjero
5	DEVCAMSAFE	Cámara seguridad

Device				
id	deviceTypeCode	codename	identifier	status
1	DEVMOB	Celular Ale	+1112345678	NULL
2	DEVMOBW	Celular Juan	+1187654321	NULL
3	DEVCAMSAFE	Cámara entrada edificio	7AAF639E-C629-4F78-AB58-260F6B89453C	ONLINE
4	DEVMOB	Celular Juan 2	+1139394812	NULL

- Escriba las siguientes queries:

1. Liste las personas que tienen al menos un dispositivo asociado.

2. Liste los dispositivos que NO tienen una persona asociada.

3. Inserte un nuevo registro en la tabla de dispositivos y asócielo a una persona ya existente.

4. Frecuentemente se hacen búsquedas por tipos de documento de la persona, proponga al menos un índice para mejorar la performance del sistema en estos casos.

Ejercicio Teórico 1 (C#)

Marque la opción correcta con una "X".

1. ¿Para que se utilizan las propiedades estáticas? ¿y las virtuales?
 - a. La clase estática permite crear múltiples instancias y luego se eliminan todas juntas por el garbage collector. Las clases virtuales se usan como plantilla y no permiten cambiar sus declaraciones de métodos.
 - b. Una clase estática es lo mismo que una clase no estática, se comportan igual salvo en cómo gestiona el framework el uso de memoria. La clase virtual son el único tipo de clases que permiten el mecanismo de herencia.
 - c. Una clase estática es básicamente lo mismo que una clase no estática, con la diferencia de que no se pueden crear instancias de una clase estática. Una clase virtual no permite sobrescribir sus miembros (métodos, propiedades, etc).
 - d. Una clase estática es básicamente lo mismo que una clase no estática, con la diferencia de que no se pueden crear instancias de una clase estática. Una clase virtual por ejemplo para un método permite que cualquier clase herede de la misma modifique su declaración.

2. ¿Qué significa que un tipo o clase es sealed?
 - a. Sealed impide que otras clases hereden de ella e impide que se invaliden métodos o propiedades virtuales.
 - b. Métodos o clases con sealed no pueden ser usados o invocados desde otro assembly.
 - c. Sealed se utiliza en métodos y propiedades se indica para forzar que sean redefinidos en las clases que lo heredan.
 - d. Sealed no deja utilizar métodos o propiedades en una clase heredada al menos que se especifique el modificador override.

3. ¿Qué distingue el managed code del unmanaged code en .NET?
 - a. El código administrado es código cuya ejecución está administrada en tiempo de ejecución, posee un control automático de memoria, seguridad, tipado, etc. El código no administrado deberá considerar todas estas cuestiones.
 - b. El código no administrado corre dentro del framework .Net, dentro de un pool de memoria especial y el código administrado es gestionado también por el framework .NET pero además es compartido por todos los lenguajes de .net
 - c. El código administrado corre dentro del framework .Net y el no administrado no corre dentro del framework .Net, pero ambos son administrados en manejo de memoria, seguridad y tipado por el sistema operativo.
 - d. El código administrado es código que posee un control automático de memoria, seguridad, tipado, etc que lo brinda el sistema operativo Windows. El código no administrado deberá considerar todas estas cuestiones.

4. ¿De qué formas se pueden utilizar los Generics de .NET?
 - a. Como definiciones de clases, interfaces o estructuras a modo de template.
 - b. Como parámetros genéricos en la definición de un método.
 - c. Como definiciones de métodos genéricos.
 - d. Todas las anteriores.

5. ¿Puede una aplicación .NET tener componentes escritos en distintos lenguajes de programación de alto nivel?
 - a. Sí, una aplicación .NET puede tener componentes escritos en cualquier lenguaje de programación de alto nivel.
 - b. Sí, siempre que el lenguaje de alto nivel se adapte a las reglas de las especificaciones CTS y CLS.
 - c. No, una aplicación .NET debe estar escrita en un único lenguaje
 - d. Sí, pero con un límite de dos lenguajes diferentes por aplicación.

Ejercicio Teórico 2 (Angular)

Marque la opción correcta con una "X".

1. ¿Cuál de los siguientes es una función built-in de Angular JS?
 - a. \$route
 - b. \$window
 - c. \$http
 - d. Todas las anteriores
2. ¿Qué directiva permite visualizar una vista para varias rutas?
 - a. ng-view
 - b. ng-model
 - c. ng-display
 - d. Ninguna de las anteriores
3. Por defecto, ¿Cómo observa un ng-model a una directiva?
 - a. Por valor
 - b. Por valor y por referencia
 - c. Por referencia
 - d. Ninguna de las anteriores

4. ¿Qué imprime el siguiente código?

```
<div>
  <p>Result: {{ "Hola Mundo" }}</p>
</div>
```

- a. Result: {{ "Hola Mundo" }}
- b. Result: Hola Mundo
- c. Result: { "Hola Mundo" }
- d. Ninguna de las anteriores

5. ¿Qué imprime el siguiente código?

```
<div>
  {{ 4/2 }}
</div>
<div id="myApp" ng-app>
  <p>{{ 5/2 }}</p>
<div>
  {{ 10/2 }}
</div>
</div>
<div>
  {{ 10/2 }}
</div>
```

- a. 2 2,5 5 5
- b. {{ 4/2 }} 2,5 5 5
- c. {{ 4/2 }} 2,5 5 {{ 10/2 }}
- d. {{ 4/2 }} 2,5 {{ 10/2 }} {{ 10/2 }}

6. ¿Cómo se pueden combinar expresiones con filters?
 - a. Usando comas: {{ expression, filter }}
 - b. Usando puntos: {{ expression . Filter }}
 - c. Usando pipes: {{ expression | filter }}
 - d. Usando barras: {{ expression / filter }}