

ESCUELA TÉCNICA DE CASTILLOS
CUARTO PRÁCTICO DE PROGRAMACIÓN III – 2019

Tema: Manejo de peticiones AJAX

Ejercicio 1: Para empezar con algo fácil, leer y anotar

- a) Explicar con sus palabras qué es una REST API y cuál es su utilidad en Programación.
- b) Explicar qué es lo que se conoce como “backend de REST” y a qué nos referimos cuando decimos “consumir servicios web”.
- c) En este nuevo patrón de trabajo, explique la diferencia entre el “front end” y el “back end” de una aplicación.
- d) Explique qué es AJAX en programación y escriba 3 ejemplos de objetos (cualesquiera, pueden ser personas, autos, dinosaurios, etc) en formato JSON.

Ejercicio 2: Aplicando a C#

El consumo de servicios web es especialmente útil cuando programamos utilizando el Framework de .NET en especial cuando quiero construir aplicaciones web en ASP, sin embargo, para nuestro caso de estudio programaremos en C# para escritorio, así que le va a tocar no sólo investigar en Internet, sino adaptar lo que encuentre a su solución particular. Se recomienda leer el artículo [siguiente](#) (blog CSharpCorner) y tomarlo como referencia.

- a) Escriba un programa que posea 2 botones, uno dirá “Listar Usuarios” y el otro “Listar Posts”. Programe los eventos correspondientes para que al hacer clic en el primer botón se guarden todos los usuarios obtenidos por una petición a la URL <https://reqres.in/api/users?page=2> mientras que si se hace click en el segundo botón, la petición será a <https://jsonplaceholder.typicode.com/posts>. En ambos casos, guarde toda la información obtenida en algún tipo de datos que considere (objeto, string, etc) y muéstrelo por pantalla.
- b) Al proyecto anterior agréguele 1 formulario, que consistan en tablas donde se muestre formateada de una manera legible y prolija la información recibida por las peticiones de usuarios o de posts, de modo que al presionar clic en el botón que corresponda, en lugar de aparecer todo el texto en formato JSON, se deserialize ese contenido y se muestre de manera amigable al usuario.
- c) Si has llegado hasta este punto, entonces estás en condiciones de aplicar **hilos** al programa. Modifique el código para que las acciones a tomar mientras se presiona cada respectivo botón (realizar la petición y mostrar el resultado en un formulario a parte) se hagan dentro de un hilo diferente para cada caso. Al finalizar las instrucciones, el hilo debe ponerse a “dormir” durante 3 segundos.

Ejercicio 3: Ya realizaron peticiones por GET, momento de probar POST, PUT y DELETE.

- a) Utilizando los recursos provistos en la web <https://reqres.in>, desarrolle un proyecto en C# donde se pida al usuario por pantalla su nombre y ocupación, y el sistema realice un **POST** a la URL de <https://reqres.in/api/users/2> con el nombre ingresado en el campo “name” y la ocupación en el campo “job”. Por último, se deben mostrar por pantalla el resultado devuelto por la API.
- b) Utilizando los recursos provistos en la web <https://reqres.in>, desarrolle un proyecto (u otra ventana en el proyecto anterior) en C# donde se pida al usuario por pantalla una ocupación, y el sistema realice un **PUT** a la URL de <https://reqres.in/api/users/2> para que a la persona con el nombre (name) “morpheus” actualice su campo “job” por la ocupación ingresada. Por último, se debe mostrar por pantalla el resultado devuelto por la API.
- c) Utilizando los recursos provistos en la web <https://reqres.in>, desarrolle un proyecto (u otra ventana en el proyecto anterior) en C# donde se pida al usuario por pantalla una ID (numérica), y el sistema realice un **DELETE** a la URL de <https://reqres.in/api/users/2> para que a la persona con la id ingresada sea **eliminada**. Por último, se debe mostrar por pantalla el resultado devuelto por la API.