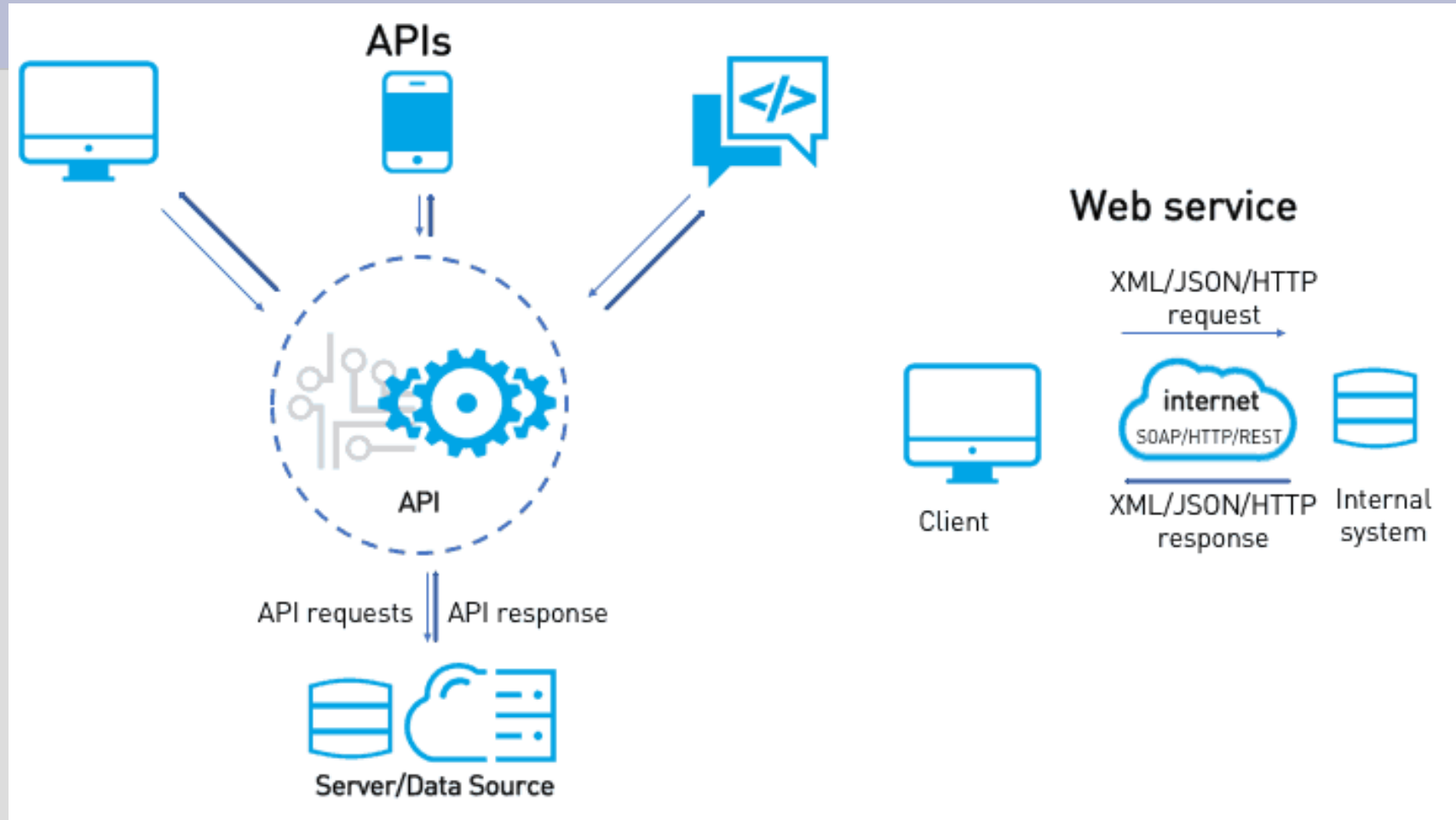


API REST – WEB SERVICES



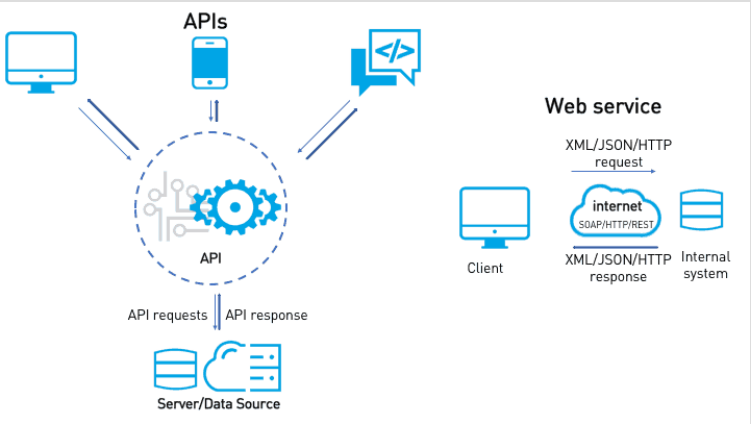
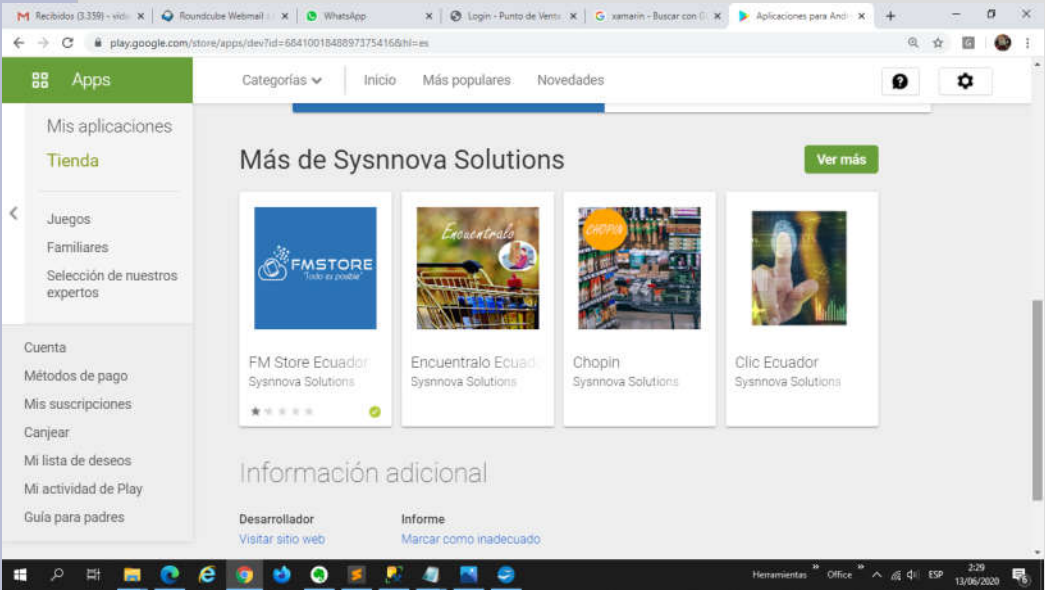
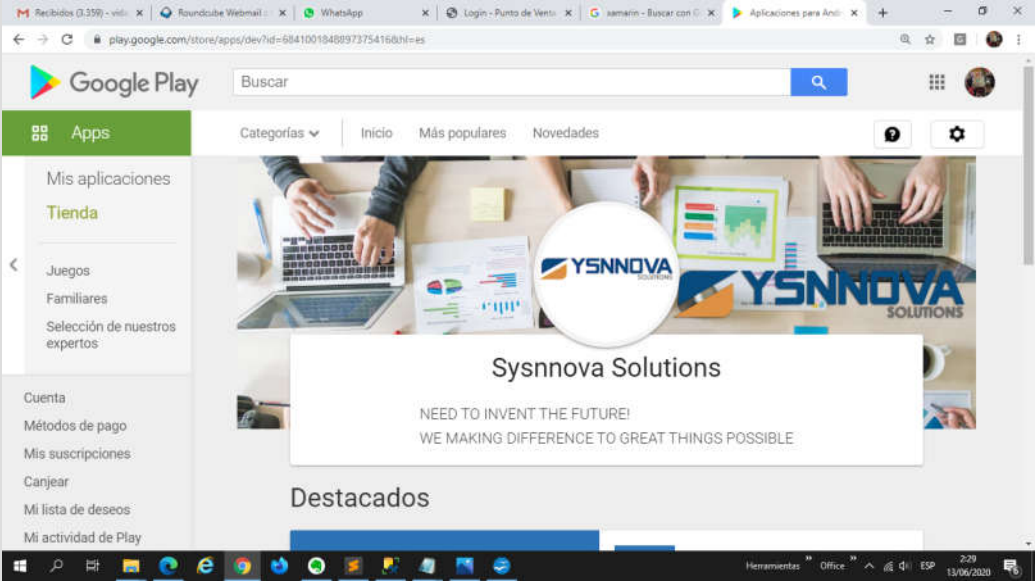


Victor Portugal

Microsoft Certified Technology Specialist (MCTS).
Certified Scrum Master (CSM).
Certified Scrum Developer (CSD).
Design Thinking Professional Certificate (DTPC)
Associate Android Developer (AAD).

Microsoft
CERTIFIED
*Technology
Specialist*





**100% PRACTIVO TALLERES API REST – WEB
SERVICES – ENTITY FRAMEWORK – NET CORE**

**INFORMIX
SQL SERVER – MY SQL**

MVC – WEB FORM – WIN FORM

APIS

API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos

El lanzamiento del nuevo sistema REST como protocolo de intercambio y manipulación de datos en los servicios de internet cambió por completo el desarrollo de *software* a partir de 2000. Ya casi toda empresa o aplicación dispone de una API REST para creación de negocio.



REST cambió por completo la ingeniería de software a partir del 2000. Este nuevo enfoque de desarrollo de proyectos y servicios web fue definido por **Roy Fielding**, el padre de la especificación HTTP y uno los referentes internacionales en todo lo relacionado con la Arquitectura de Redes, en su disertación '[Architectural Styles and the Design of Network-based Software Architectures](#)'. En el campo de las APIs, REST (Representational State Transfer- [Transferencia de Estado Representacional](#)) es, a día de hoy, el alfa y omega del desarrollo de servicios de aplicaciones.

En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios profesionales a partir de ese software. Twitter, YouTube, los sistemas de identificación con Facebook... hay cientos de empresas que generan negocio gracias a REST y las APIs REST. Sin ellas, todo el crecimiento en horizontal sería prácticamente imposible. Esto es así porque REST es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet.

Buscando una definición sencilla, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad pero también mucha complejidad. A veces es preferible **una solución más sencilla de manipulación de datos como REST**.

Características de REST

- **Protocolo cliente/servidor sin estado:** cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como **protocolo cliente-caché-servidor sin estado:** existe la posibilidad de definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro **la misma respuesta para peticiones idénticas**. De todas formas, que exista la posibilidad no significa que sea lo más recomendable.
- Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: **POST** (crear), **GET** (leer y consultar), **PUT** (editar) y **DELETE** (eliminar).

- **Los objetos en REST siempre se manipulan a partir de la URI.** Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.
- **Interfaz uniforme:** para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.

Ventajas que ofrece REST para el desarrollo

1. **Separación entre el cliente y el servidor:** el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos. Eso tiene algunas ventajas cuando se hacen desarrollos. Por ejemplo, mejora la portabilidad de la interfaz a otro tipo de plataformas, aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente.
2. **Visibilidad, fiabilidad y escalabilidad.** La separación entre cliente y servidor tiene una ventaja evidente y es que cualquier equipo de desarrollo puede escalar el producto sin excesivos problemas. Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta. Esta separación facilita tener en servidores distintos el *front* y el *back* y eso convierte a las aplicaciones en productos más flexibles a la hora de trabajar.

3. **La API REST siempre es independiente del tipo de plataformas o lenguajes:** la API REST siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Con una API REST se pueden tener servidores PHP, Java, Python o Node.js. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente XML o JSON.

Cursos - Google Drive

localhost/todoitems/api/todoitems

localhost/todoitems/api/todoitems 90%

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

```
{
  "0": {
    "id": "6bb8a868-dba1-4f1a-93b7-24ebce87e243",
    "name": "Learn app development",
    "notes": "Xamarin University",
    "done": true
  },
  "1": {
    "id": "6bb8a868-d93b7-94ebce87e243",
    "name": "Api development",
    "notes": "MA University",
    "done": false
  },
  "2": {
    "id": "CC8a968-d93b7-94ebce87e290",
    "name": "Publish Apps in Market",
    "notes": "Sipecom",
    "done": false
  }
}
```

Herramientas » Office » 8:46 18/07/2020