

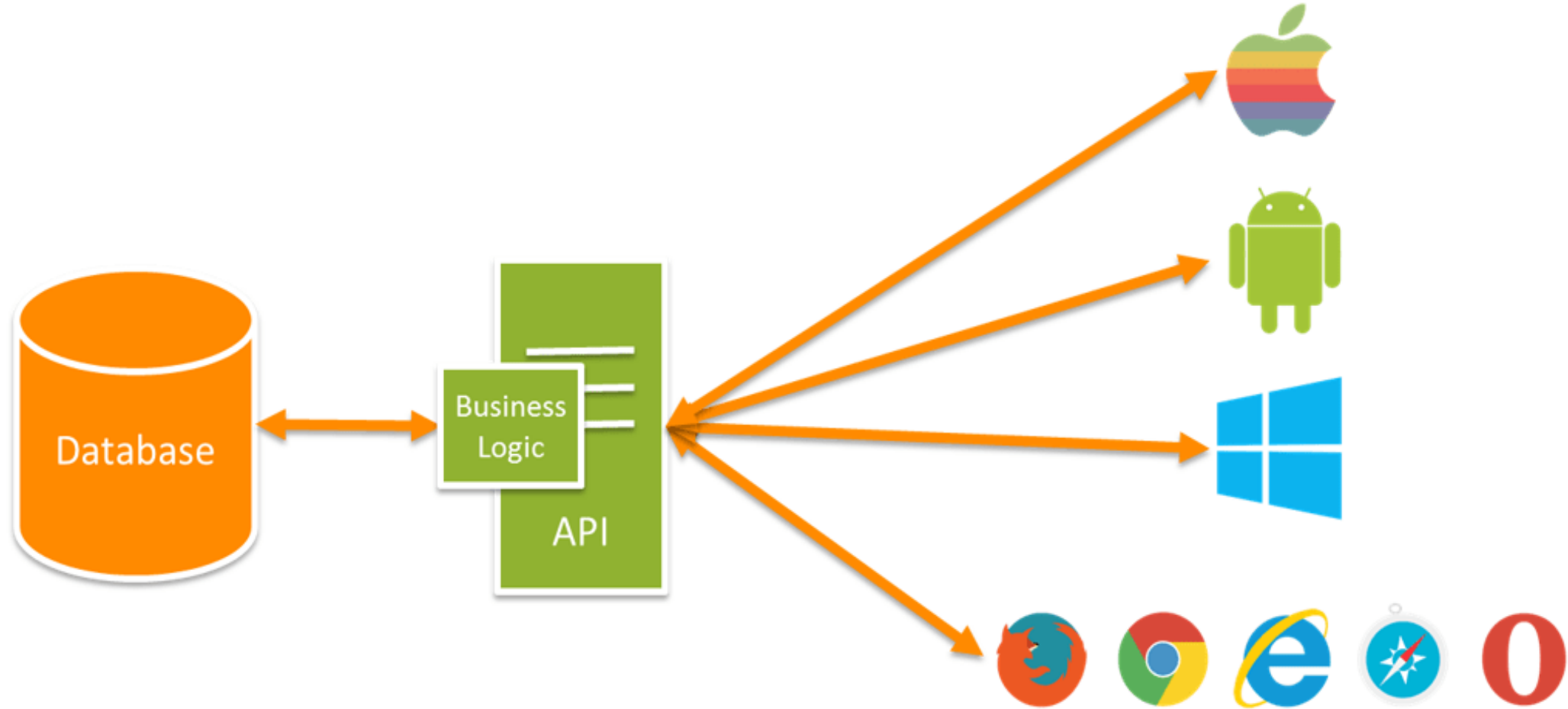


Web Api - Rest



C#
Csharp



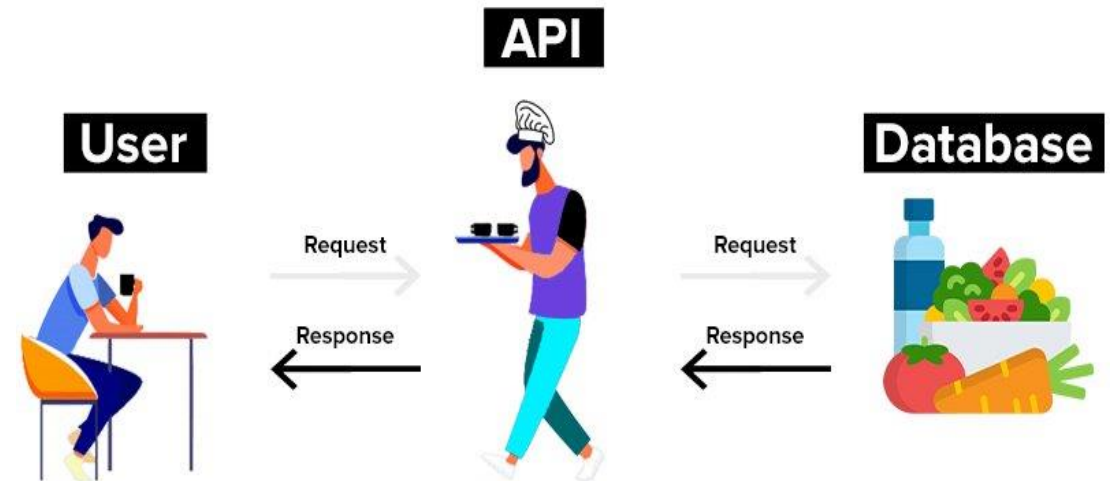






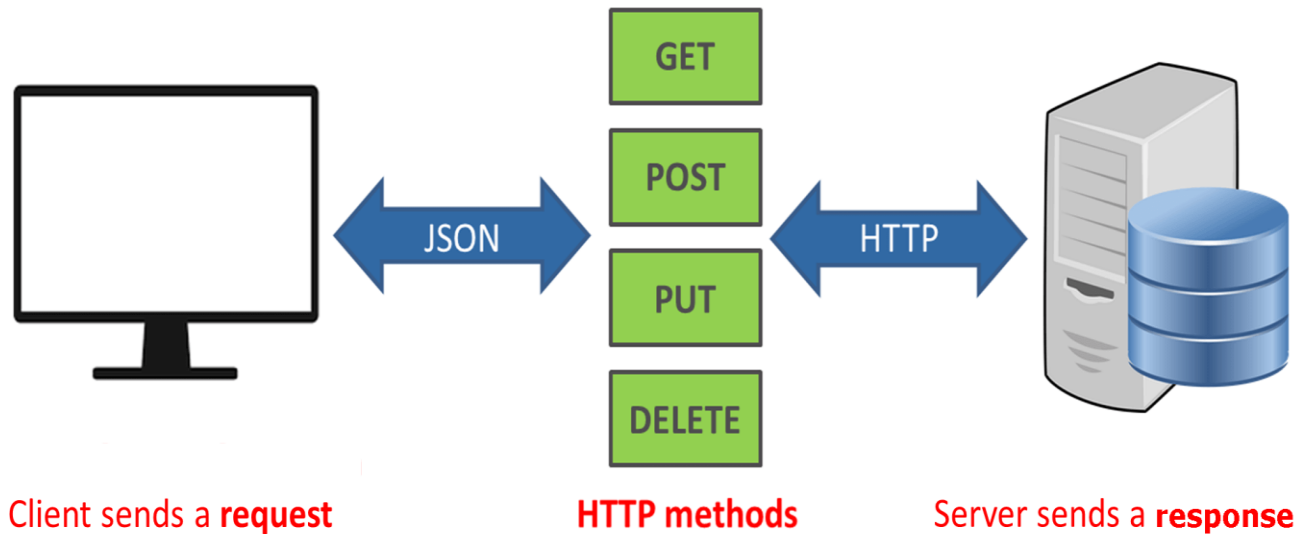
Qué es una API

El término API es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.





HTTP define métodos que son utilizados para diferentes circunstancias, por lo que trataremos de listar lo más relevantes y más utilizamos en la construcción de servicios REST, los métodos son los siguientes:



GET: Es utilizado únicamente para consultar información al servidor, muy parecidos a realizar un SELECT a la base de datos. No soporta el envío del payload

POST: Es utilizado para solicitar la creación de un nuevo registro, es decir, algo que no existía previamente, es decir, es equivalente a realizar un INSERT en la base de datos. Soporta el envío del payload.

PUT: Se utiliza para actualizar por completo un registro existente, es decir, es parecido a realizar un UPDATE a la base de datos. Soporta el envío del payload.

PATCH: Este método es similar al método PUT, pues permite actualizar un registro existente, sin embargo, este se utiliza cuando actualizar solo un fragmento del registro y no en su totalidad, es equivalente a realizar un UPDATE a la base de datos. Soporta el envío del payload

DELETE: Este método se utiliza para eliminar un registro existente, es similar a DELETE a la base de datos. No soporta el envío del payload.



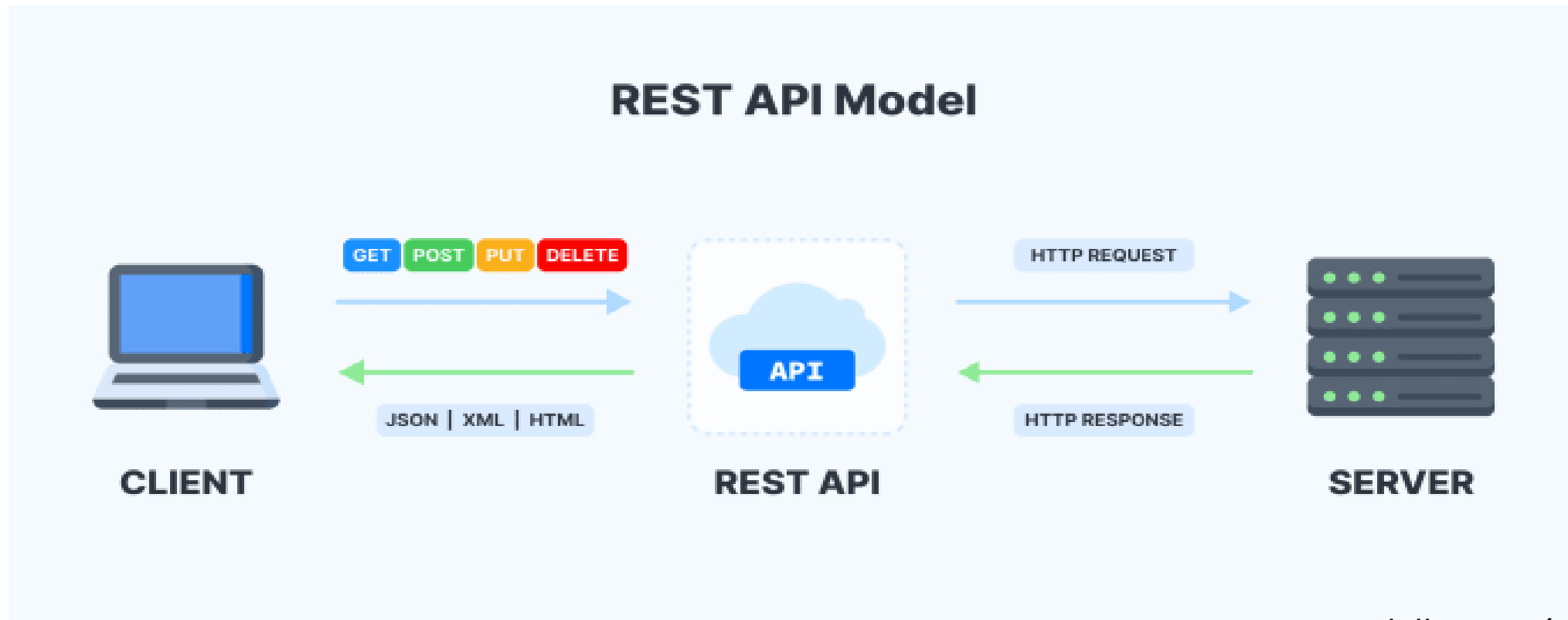
Métodos HTTP





Realizaremos un ejemplo práctico:

API Rest CRUD con el lenguaje de programación C# y SQL Server



Instructor: Adalberto Cárcamo A.



1. Creación de la base de datos:

```
CREATE DATABASE DB_API_DATOS
```

```
USE DB_API_DATOS
```



BD



2. Creamos la siguiente tabla de usuario:

```
create table USUARIO(  
  IdU varchar(15) primary key,  
  Nombres varchar(80),  
  Telefono varchar(60),  
  Correo varchar(40),  
  Ciudad varchar(60),  
  FechaIngreso datetime default getdate()  
)
```



3. Ingresamos 5 registros de prueba en la tabla de usuario:



```
INSERT INTO USUARIO (IdU, Nombres, Telefono, Correo, Ciudad)  
VALUES ( '545454', 'Sara', 3445654321, 'saracar@gmail.com', 'Cartagena' )
```



4. Creamos la capa de datos :

Iniciamos con los procedimientos almacenados

- a. Registrar
- b. Actualizar
- c. Eliminar
- d. Consultar
- e. Listar





a. Registrar Usuario



```
create procedure usp_registrar(  
@idu varchar(15),  
@nombres varchar(80),  
@telefono varchar(60),  
@correo varchar(40),  
@ciudad varchar(60)  
)  
as  
begin  
insert into USUARIO(IdU,Nombres,Telefono,Correo,Ciudad)  
values  
(  
@idu,  
@nombres,  
@telefono,  
@correo,  
@ciudad  
)  
end
```



b. Actualizar Usuario



```
create procedure usp_actualizar (  
    @idu varchar(15),  
    @nombres varchar(80),  
    @telefono varchar(60),  
    @correo varchar(40),  
    @ciudad varchar(60)  
)  
as  
begin  
    update USUARIO set  
        Nombres = @nombres,  
        Telefono = @telefono,  
        Correo = @correo,  
        Ciudad = @ciudad,  
    where IdU = @idu  
end
```



c. Eliminar Usuario



```
create procedure usp_eliminar(  
@idu varchar(15)  
)  
as  
begin  
  
delete from usuario where IdU = @idusuario  
  
end
```



d. Consultar Usuario



```
create procedure usp_obtener  
    @idu varchar(15)  
  
as  
begin  
    select * from usuario where IdU = @idu  
end
```



e. Listar Usuario



```
create procedure usp_listar
as
begin

select * from usuario
end
```