

Notaciones literales para la comunicación y almacenamiento de datos

XML y JSON



Lenguaje de marcas extensibles *Extensible Markup Language* XML

Notación que permite **definir etiquetas personalizadas** que describen y organizan los **datos** para su **comunicación** y **almacenamiento**

Estructura de un XML

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Mensaje>
```

```
  <Remitente>
```

```
    <Nombre>Nombre</Nombre>
```

```
    <Mail>Correo</Mail>
```

```
  </Remitente>
```

```
  <Destinatario>
```

```
    <Nombre>Nombre</Nombre>
```

```
    <Mail>Correo</Mail>
```

```
  </Destinatario>
```

```
  <Texto>
```

```
    Contenido del mensaje.
```

```
  </Texto>
```

```
</Mensaje>
```



Notación de objetos de JavaScript

JavaScript Object Notation **JSON**

Notación literal (subconjunto) de **objetos** de **JavaScript**,
para la **comunicación** y **almacenamiento** de **datos**

Ejemplo de definición de un objeto JSON

- JSON puede ser leído de forma fácil por cualquier lenguaje de programación.
- Fácil manejo en JavaScript.
- Uso para el intercambio de información entre distintas tecnologías.

```
{ "nombre": "Fulgencio" }
```

```
{ "cuenta": 12345678, "nombre": "Fulgencio" }
```

```
{  
  "alumnos": [  
    { "cuenta": 12345678, "nombre": "Fulgencio" },  
    { "cuenta": 21256337, "nombre": "Liborio" },  
    { "cuenta": 41235678, "nombre": "Sinforoso" }  
  ]  
}
```

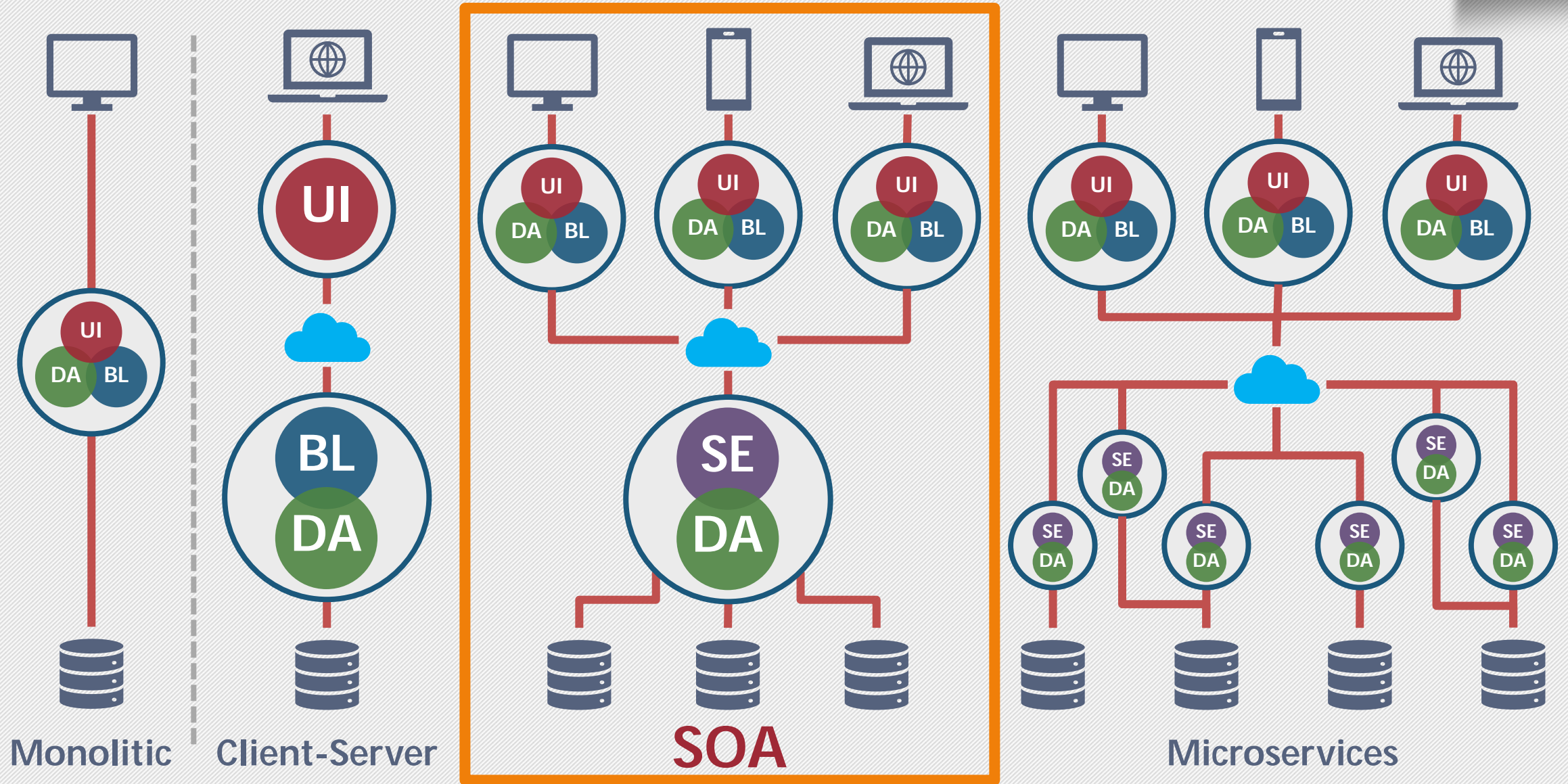


Arquitectura Orientada a Servicios *Service Oriented Architecture* SOA

Diseño arquitectónico basado en la representación lógica de **reglas de negocios** que acepta **peticiones** y devuelve **respuestas** mediante una **interfaz** bien definida

Estilos arquitectónicos

Arquitecturas





Interfaz de Programación de Aplicaciones *Application Programming Interface* **API**

Es un conjunto de rutinas que provee **acceso** a **características** de bajo nivel o **propietarias** de un determinado **software**

Web Services y API Web

SOAP y REST



Protocolo de Acceso Simple a Objeto *Simple Object Access Protocol SOAP*

Protocolo de mensajería que puede ser utilizado para formar **protocolos más complejos y completos**, y cuyos mensajes en **XML** son **independientes** y pueden ser transportados por **protocolos de Internet** (SMTP, MIME y HTTP)

SOAP

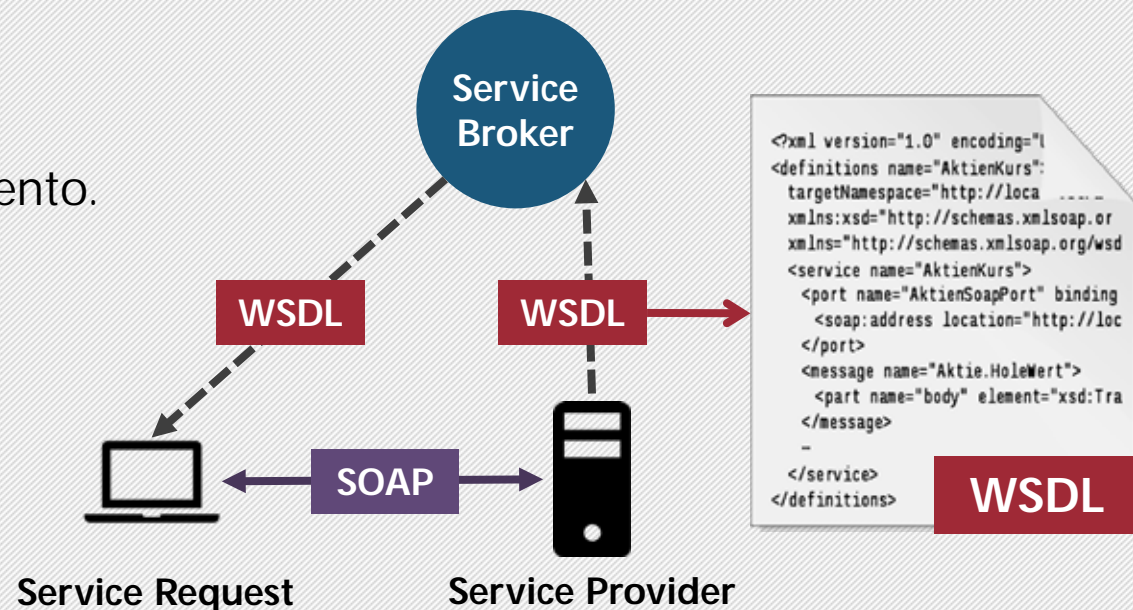
Básicamente SOAP es un paradigma de mensajería **de una dirección sin estado**, ofreciendo un framework de mensajería básica en el cual los **Web Services** se pueden construir.

Ventajas:

- Debido al uso de XML permite invocar procedimientos remotos de muchos lenguajes.
- Al utilizar HTTP es fácilmente escalable, además de ser permitido por los cortafuegos.
- Es posible transmitirlo por cualquier protocolo de transporte como SMTP.

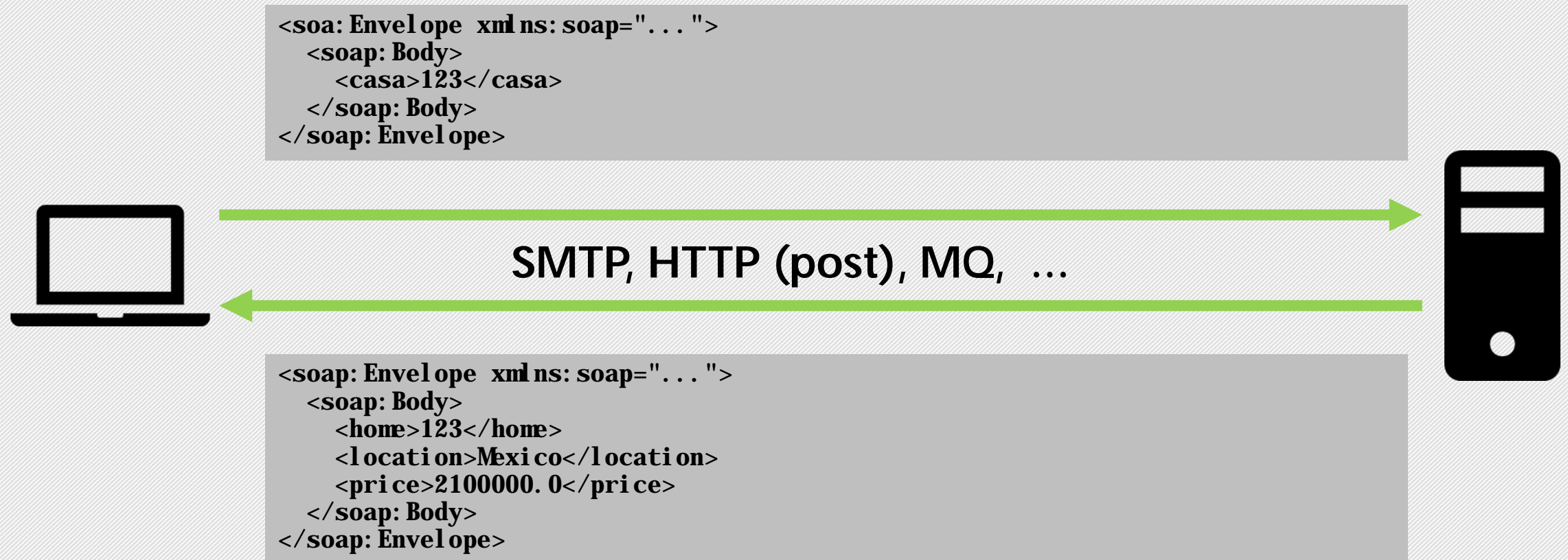
Desventajas:

- Debido al uso de XML es considerablemente más lento.
- Depende del **WSDL** (Web Services Description Language).
- Al contrario que Java, PHP o Python ciertos lenguajes no ofrecen un apoyo adecuado para su uso ya sea a nivel de integración o de soporte IDE.



Web Services

SOAP



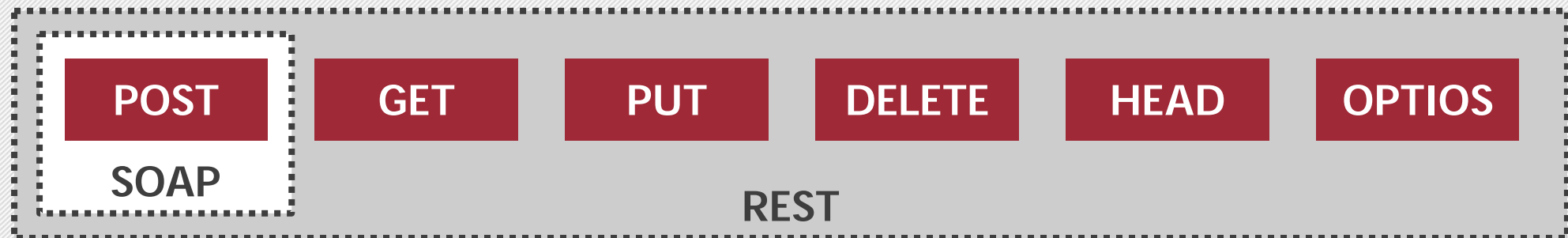


Transferencia de estado representacional *Representational State Transfer* **REST**

Estilo de arquitectura ligero y con una **interface uniforme** que trabaja sobre **HTTP** para la interoperabilidad de **datos** y las **operaciones** sobre ellos

REST

- Fácil de entender y que permite un **uso eficiente del ancho de banda**.
- A diferencia de SOAP es **más ligero**, permite **múltiples** tecnologías de comunicación de datos (XML, JSON, etcétera)
- No **almacena estado**; las lecturas pueden guardarse en caché para mejorar el rendimiento y escalabilidad.
- Muy usado cuando se requiere muchos mensajes de ida y vuelta, y en aplicaciones que dejan de responder ya que permite activar procesos de re-intento.
- Cuenta con una interface uniforme lo que lo permite una mayor independencia





API Web REST

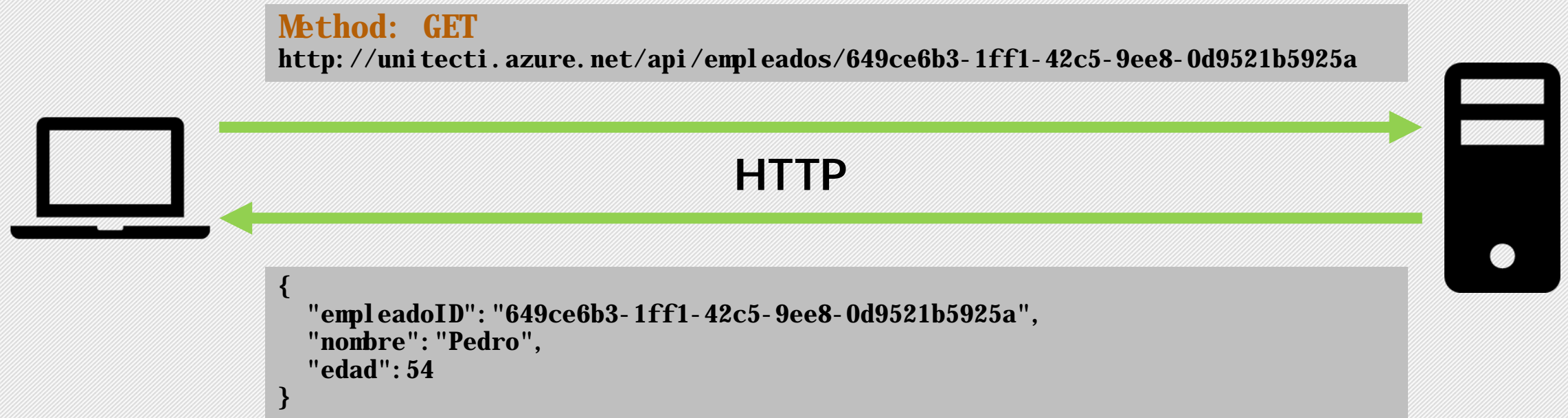


Tabla comparativa

Ventajas SOAP	Ventajas REST
Pocas operaciones con muchos recursos	Muchas operaciones con pocos recursos
Envió solo por XML	Envió con cualquier formato
SMTP, HTTP (post), MQ	HTTP (get, post, put, delete, options, etc)
Tipado fuerte, XML Schema	XML auto descriptivo
Síncrono y Asíncrono	Síncrono
WS SECURITY	HTTPS
Comunicación origen a destino seguro	Comunicación punto a punto y segura
Finanzas, Telecomunicaciones, Portales de pago	Social media, Web Chat, Mobile

Interfaz uniforme en REST

URI	POST <i>Crear</i>	GET <i>Obtener</i>	PUT <i>Modificar</i>	DELETE <i>Borrar</i>
http://localhost/api/casas	Error	Devuelve todas las casas	Modifica todas las casas *	Borra todas las casas
http://localhost/api/casas/123	Crea una casa	Devuelve la casa	Modifica la casa	Borra la casa