

15_aplicaciones_y_servicios_web

October 15, 2020

1 Aplicaciones y servicios web.

1.1 Aplicación web (“web app”).

Una aplicación web por lo general está enfocada a realizar transacciones que son accesibles a un usuario final mediante la interfaz de un navegador web. Esta interfaz, también conocida como “front-end” es el medio de interacción entre el usuario y dicha aplicación.

1.2 Servicios web (“web services”).

Un servicio web está enfocado a la realización de transacciones en la que el énfasis se hace en el intercambio de datos entre el cliente y el servidor mediante al acceso a “endpoints”.

1.2.1 Puntos de acceso (“endpoints”).

Un “endpoint” es una *URL* que expone un servicio web, dicho servicio puede ser consumido por un cliente (no necesariamente un navegador web) en caso de que cuente con la identidad y los permisos correctos.

1.2.2 Interfaces de programación de aplicaciones (API).

Las API son un conjunto de endpoints que exponen diversas funcionalidades de un servicio web, las cuales pueden ser utilizadas para el desarrollo de aplicaciones web.

1.2.3 Aplicaciones web desde el front-end.

1.3 El patrón Modelo Vista Controlador (*MVC*).

Este patrón de diseño de aplicaciones web consiste en separar los datos a los que accede una aplicación de las interfaces de dicha aplicación.

El patrón *MVC* se volvió muy popular en el desarrollo de aplicaciones web particularmente con su implementación por medio de *Ruby on Rails*.

1.3.1 *Modelo*.

El Modelo corresponde a la estructura de datos de la aplicación, la cual por lo general es gestionada por una base de datos y puede ser ligada a una estructura orientada a objetos mediante un Modelo Relacional de Objetos (*ORM* por sus siglas en inglés).

El estado de una aplicación se refleja en el modelo, mientras que las vistas y los controladores son interfaces que permiten interactuar con el modelo.

1.3.2 *Vista.*

La vista es el modo en el que el estado del modelo es presentado al usuario.

1.3.3 *Controlador.*

Los controladores son gestores de eventos los cuales tienen la capacidad de: * Interactuar con el estado del modelo. * Interactuar con lo que despliega vista.

1.4 Arquitecturas y protocolos de servicios web.

1.4.1 *SOAP.*

Protocolo Simple de Acceso a Objetos, o *SOAP* es un protocolo que define la forma en la que dos objetos pueden comunicarse mediante el intercambio de datos en formato *XML*. La especificación de SOAP es publicada por la W3C.

Las arquitecturas de servicios web basadas en *SOAP* definen a detalle el proceso de construcción y procesamiento de mensajes que permiten una comunicación bidireccional entre dos extremos.

1.4.2 *RESTful o REST.*

REST corresponde a las siglas en inglés de *Representación de Estado Transaccional* y fue propuesta por primera vez en la [tesis doctoral de Roy Fielding](#) como una serie de reglas para aprovechar los métodos de *HTTP* con la finalidad de crear servicios web ligeros y rápidos. Los servicios basados en *REST* pueden transmitir recursos en diversos formatos, incluyendo *XML* y *JSON*.

Referencia: <https://www.restapitutorial.com/>

1.5 *GraphQL.*

GraphQL es un lenguaje de consultas creado por Facebook con la finalidad de diseñar APIs que permiten optimizar la calidad y detalle de la transferencia de datos.

GraphQL ha sido implantado en [varios lenguajes de programación](#) y ofrece [diversos clientes](#) capaces de consumir las *APIs* creadas con este lenguaje.

1.6 Formatos de serialización y transferencia de datos.

1.6.1 *XML.*

El *Lenguaje Extensible de Etiquetas XML* es una especificación de la W3C basada en *SGML* diseñado para serializar y transmitir datos independientemente de los medios de transporte de datos. Es un meta-lenguaje que permite definir formatos derivados mediante el uso de:

- *DTD.*
- *XML Schema.*

Los derivados de *XML* permiten definir estructuras e incluso tipos de datos con propósitos específicos.

La especificación también define lenguajes de consulta que permiten extraer información de documentos *XML*.

- *XPATH*, el cual es un lenguaje que permite realizar búsquedas dentro de una estructura *XML* por medio de rutas a partir de la raíz del documento.
- *XQuery* es un lenguaje que permite hacer consultas en un documento *XML* mediante una sintaxis similar a *SQL*.

Incluso es posible transformar un documento con una estructura específica de *XML* en otra mediante *XSLT*.

1.6.2 *JSON*.

JSON es un acrónimo de “*Javascript Object Notation*” y corresponde en un principio al modo en que se definen objetos en *Javascript* según las especificaciones previas a *ECMAScript* 5. Sin embargo, la notación es tan simple y práctica que ha sido adoptada como un formato para la serialización y transmisión de datos.

Existen incluso lenguajes derivados de *JSON* tales como *YAML*.

Es posible consultar más sobre *JSON* en la siguiente liga:

<https://www.json.org/json-en.html>

1.7 Mejores prácticas para el desarrollo de aplicaciones SaaS.

La metodología “The Twelve-Factor App” enumera y describe 12 elementos críticos que deben de ser tomados en cuenta cuando se desarrollan aplicaciones que ofrecerán servicios en línea.

1. Código base (Codebase)
2. Dependencias
3. Configuraciones
4. Backing services
5. Construir, desplegar, ejecutar
6. Procesos
7. Asignación de puertos
8. Concurrencia
9. Desechabilidad
10. Paridad en desarrollo y producción
11. Historiales
12. Administración de procesos

El documento completo puede ser consultado en <https://12factor.net/es/>

Esta obra está bajo una Licencia Creative Commons Atribución 4.0 Internacional.

© José Luis Chiquete Valdivieso. 2020.