SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR
TECNOLÓGICA

INSTITUTO TECNOLÓGICO
DE OAXACA

INSTITUTO TECNOLÓGICO DE OAXACA

ING. EN SISTEMAS COMPUTACIONALES

DISEÑO DE SOFTWARE ORIENTADO A SERVICIOS.

TEMA:

INVESTIGACIÓN I.

HORA:

7:00 - 8:00 / I13

CATEDRÁTICO:

L.I. Iván Adán.

PRESENTA:

- **Zaragoza Farrera Luis Angel.**

1968-2018 ANIVERSARIO INSTITUTO TECNOLÓGICO DE OAXACA

A. ¿Qué es la programación síncrona?

Los lenguajes de programación asíncronos se basan en llamadas que pueden ser cumplidas ahora o en un futuro. Es decir, las variables pueden ser llenadas o asignadas en cualquier momento de la ejecución del programa.

Muchos de los lenguajes de programación populares se basan en procesos síncronos. Es decir, una orden solo se puede ejecutar luego que se ejecuta la anterior.

Tradicionalmente, el modelo de ejecución que utilizan la mayoría de los lenguajes y paradigmas de programación se corresponde con un tratamiento secuencial del cuerpo del programa. Un programa es entendido como una secuencia de instrucciones que se ejecutan ordenadamente y donde la ejecución de cada operación no da comienzo hasta que no termina la anterior.

B. ¿Qué es la programación asíncrona?

La programación asíncrona establece la posibilidad de hacer que algunas operaciones devuelvan el control al programa llamante antes de que hayan terminado mientras siguen operando en segundo plano. Esto agiliza el proceso de ejecución y en general permite aumentar la escalabilidad, pero complica el razonamiento sobre el programa.

Un modelo asincrónico permite que sucedan múltiples cosas al mismo tiempo. Al iniciar una acción, el programa continúa ejecutándose. Cuando la acción termina, el programa es informado y tiene acceso al resultado (por ejemplo, los datos leídos desde el disco).

Programación asíncrona orientada al framework de JavaScript NodeJS se manejan diferentes tipos de modelos, que son los siguientes:

- **Modelo de paso de continuadores**

Es el modelo de asincronía más utilizado dentro Node JS. Cada función recibe información acerca de cómo debe tratar el resultado –de éxito o error– de cada operación. Requiere orden superior.

- **Modelo de eventos**

Se utiliza una arquitectura dirigida por eventos que permite a las operaciones no bloqueantes informar de su terminación mediante señales de éxito o fracaso. Requiere correlación para sincronizar.

- **Modelo de promesas**

Se razona con los valores de retorno de las operaciones no bloqueantes de manera independiente del momento del tiempo en que dichos valores –de éxito o fallo– se obtengan.

- **Modelo de generadores**

Se utilizan generadores para devolver temporalmente el control al programa llamante y retornar en un momento posterior a la rutina restaurando el estado en el punto que se abandonó su ejecución.

C. API’s

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones.

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

En resumen, las API le permiten habilitar el acceso a sus recursos y, al mismo tiempo, mantener la seguridad y el control. Cómo habilitar el acceso y a quiénes depende de usted. La seguridad de las API tiene que ver con que se gestionen bien. Para conectarse a las API y crear aplicaciones que utilicen los datos o las funciones que estas ofrecen, se puede utilizar una plataforma de integración distribuida que conecte todos los elementos, incluidos los sistemas heredados y el Internet de las cosas (IoT).

Existen tres enfoques respecto a las políticas de las versiones de las API.

- **Privado**

Las API solo se pueden usar internamente, así que las empresas tienen un mayor control sobre ellas. Esto le da a las empresas un mayor control sobre sus API.

- **De Partners**

Las API se comparten con partners empresariales específicos, lo cual puede ofrecer flujos de ingresos adicionales, sin comprometer la calidad. Esto puede proporcionar flujos de ingreso adicionales, sin comprometer la calidad.

- **Público**

Todos tienen acceso a las API, así que otras empresas pueden desarrollar API que interactúen con las de usted y así convertirse en una fuente de innovaciones. Esto permite que terceros desarrollen aplicaciones que interactúan con su API, y puede ser un recurso para innovar.

Una API (siglas de ‘Application Programming Interface’) es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

Las API pueden servir para comunicarse con el sistema operativo (WinAPI), con bases de datos (DBMS) o con protocolos de comunicaciones (Jabber/XMPP). En los últimos años, por supuesto, se han sumado múltiples redes sociales (Twitter, Facebook, Youtube, Flickr, LinkedIn, etc) y otras plataformas online (Google Maps, WordPress…), lo que ha convertido el social media marketing es algo más sencillo, más rastreable y, por tanto, más rentable.

Las API son valiosas, ante todo, porque permiten hacer uso de funciones ya existentes en otro software (o de la infraestructura ya existente en otras plataformas) para no estar reinventando la rueda constantemente, reutilizando así código que se sabe que está probado y que funciona correctamente.

D. ¿Qué es un servicio web?

El término Web Services describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. XML es usado para describir los datos, SOAP se ocupa para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa para conocer cuales son los servicios disponibles. Uno de los usos principales es permitir la comunicación entre las empresas y entre las empresas y sus clientes. Los Web Services permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información.

A diferencia de los modelos Cliente/Servidor, tales como un servidor de paginas Web, los Web Services no proveen al usuario una interfaz gráfica (GUI). En vez de ello, los Web Services comparten la lógica del negocio, los datos y los procesos, por medio de una interfaz de programas a través de la red. Es decir conectan programas, por tanto son programas que no interactúan directamente con los usuarios. Los desarrolladores pueden por consiguiente agregar a los Web Services la interfaz para usuarios, por ejemplo mediante una pagina Web o un programa ejecutable, tal de entregarle a los usuarios un funcionalidad específica que provee un determinado Web Service.

ESTANDARES EN WEB SERVICES.

- Web Services Protocol Stack: conjunto de servicios y protocolos de los servicios web.
- XML (Extensible Markup Language): formato estándar para los datos que se vayan a intercambiar.
- SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call): protocolos sobre los que se establece el intercambio.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), o Simple Mail Transfer Protocol (SMTP).
- WSDL (Web Services Description Language): es el lenguaje de la interfaz pública para los servicios web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios web.
- UDDI (Universal Description, Discovery and Integration): protocolo para publicar la información de los servicios web. Permite comprobar qué servicios web están disponibles.
- WS-Security (Web Service Security): protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.
- REST (Representational State Transfer): arquitectura que, haciendo uso del protocolo HTTP, proporciona una API que utiliza cada uno de sus métodos (GET, POST, PUT, DELETE, etcétera) para poder realizar diferentes operaciones entre la aplicación que ofrece el servicio web y el cliente.
- GraphQL, arquitectura alternativa a REST.

SOAP

SOAP es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir. Este protocolo está basado en XML y se conforma de tres partes:

- Sobre (envelope): el cual define qué hay en el mensaje y cómo procesarlo.
- Conjunto de reglas de codificación para expresar instancias de tipos de datos.
- La Convención para representar llamadas a procedimientos y respuestas.

El protocolo SOAP tiene tres características principales:

- Extensibilidad (seguridad y WS-routing son extensiones aplicadas en el desarrollo).
- Neutralidad (bajo protocolo de transporte TCP puede ser utilizado sobre cualquier protocolo de aplicación como HTTP, SMTP o JMS).
- Independencia (permite cualquier modelo de programación).

WSDL

WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

El WSDL nos permite tener una descripción de un servicio web. Especifica la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar.

La estructura del WSDL tiene los siguientes elementos:

- **Tipos de datos**

<types>: Esta sección define los tipos de datos usados en los mensajes. Se utilizan los tipos definidos en la especificación de esquemas XML.

- **Mensajes**

<message>: Aquí definimos los elementos de mensaje. Cada mensaje puede consistir en una serie de partes lógicas. Las partes pueden ser de cualquiera de los tipos definidos en la sección anterior.

- **Tipos de puerto**

<portType>: Con este apartado definimos las operaciones permitidas y los mensajes intercambiados en el Servicio.

- **Bindings**

<binding>: Especificamos los protocolos de comunicación usados.

- **Servicios**

<service>: Conjunto de puertos y dirección de los mismos. Esta parte final hace referencia a lo aportado por las secciones anteriores.

Con estos elementos no sabemos qué hace un servicio pero sí disponemos de la información necesaria para interactuar con él (funciones, mensajes de entrada/salida, protocolos...)

E. ¿Qué es el desarrollo web?

Desarrollo web es un término que define la creación de sitios web para Internet o una intranet. Para conseguirlo se hace uso de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador web a fin de realizar determinadas tareas o mostrar información.

Aunque desarrollo web puede ser un ámbito muy específico del desarrollo en general, el ecosistema de tecnologías, lenguajes y herramientas disponibles para la creación web es enorme. Para poder clasificar o dividir las áreas del desarrollo web se han acuñado un par de términos:

- Front-end: Es el desarrollo web en el ámbito del cliente, es decir, en el navegador web. Las tecnologías y lenguajes principales son HTML, CSS y Javascript.
- Back-end: Es el desarrollo web en el ámbito del servidor, donde las tecnologías y lenguajes son directamente relacionadas con el sistema operativo del servidor de la página web. Lenguajes y tecnologías habituales para Back-end son PHP, NodeJS, Python, .NET, e incluye también las bases de datos, como MySQL, PostgreSQL, SQL Server.

F. Laravel

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP.

La mayor parte de su estructura está formada por dependencias, especialmente de Symfony, lo que implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.