**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| PROGRAMA DE FORMACIÓN | ANÁLISIS Y DESARROLLO DE SOFTWARE |
| --- | --- |

| COMPETENCIA | 220501096 - Desarrollar la solución de *software* de acuerdo con el diseño y metodologías de desarrollo. | RESULTADOS DE APRENDIZAJE | 220501096-04 Codificar el *software* de acuerdo con el diseño establecido. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | 34 |
| --- | --- |
| NOMBRE DEL COMPONENTE FORMATIVO | Servicios web con PHP |
| BREVE DESCRIPCIÓN | Los servicios web, también conocidos como *web* *services*, son un conjunto de protocolos que facilitan la comunicación entre dispositivos, permitiendo intercambiar información: “datos”. Un servicio web tiene una interfaz que oculta los detalles de implementación, para que se pueda utilizar independientemente de la plataforma de *hardware* o *software* en la que se implementa, e independientemente del lenguaje de programación en el que está escrito. |
| PALABRAS CLAVE | servicios, WEB, PHP, aplicaciones, XML, SOAP, WSDL, UDDI |

| ÁREA OCUPACIONAL | 2 - Ciencias naturales, aplicadas y relacionadas |
| --- | --- |
| IDIOMA | Español |

1. **Tabla de contenidos**

**Introducción**

1. **Métodos o funciones**
2. **Clases y objetos**
3. **Integración de aplicaciones (XML, SOAP, WSDL y UDDI)**
   1. XML “Extensible Markup Language”
   2. SOAP “Simple Object Access Protocol”
   3. WSDL “Lenguaje de Descripción de Servicios Web”
   4. UDDI “Universal Description, Discovery, and Integration”
   5. REST “Representational State Transfer”

3.6 JASON “JavaScript Object Notation”

1. **Creación de servicios**
2. **Conexiones a SQL**
3. **Servicios en PHP**
4. **Aplicación SOAP UI**

7.1 Instalación y configuración

7.2 Uso de la herramienta SOAP UI

**Síntesis**

1. **Desarrollo de contenidos**

**Introducción**

Estimado aprendiz, bienvenido al componente formativo “Servicios web con PHP”. Para iniciar, visualice el siguiente video y conozca sobre el contenido a estudiar.



Para la elaboración de este componente, se abordaron varios autores conocidos en **servicios web con PHP**, de quienes se han citado y referenciado conceptos y ejemplos para los fines educativos de esta materia, en el entendido de que el conocimiento es social y, por lo tanto, es para ser usado por quienes necesitan adquirirlo. Se espera que este documento sea útil para todos aquellos, aprendices y lectores en general, que estén interesados en acercarse a asuntos básicos de **análisis y desarrollo de *software***.

**1. Métodos o funciones**

De acuerdo con Gonzáles y Pelissier (2002), los métodos o funciones permiten segmentar el trabajo que hace un programa en subtareas o tareas más pequeñas, enfocadas en un fin específico, y se pueden utilizar cuantas veces se necesite; estas están separadas del programa principal, pero aportan a su objetivo.

El uso de funciones da la capacidad de agrupar varias instrucciones de código bajo un solo nombre, y con el nombre de la función, es posible invocarlo las veces que se necesite, permitiendo usar esta función las veces que sean necesarias.

| megáfono1 con relleno sólidoEjemplo 1:  La función se define con la siguiente sintaxis:  *Nota:* el símbolo “//” quiere decir que no hace parte del código, se lo utiliza para hacer comentarios sobre el código, por lo cual se pone en color azul.   | 1  2  3  4  5  6  7  8  9 | <?php  function media\_aritmetica($a, $b){ //Aquí definimos la función y la llamamos media\_arimetica y las variables a y b  $media=($a+$b)/2; //escribimos la fórmula de la media, en este caso de 2 números divididos entre 2  return $media; // aquí la función retorna el valor de la media  }  echo media\_aritmetica(7,9),"<br>"; // aquí ingresamos los números, en este caso 7 y 9  echo media\_aritmetica(300,500),"<br>"; ?> //aquí ingresamos otros dos números 300 y 500 | | --- | --- |   La función arroja como resultados los números 8 y 400.  En el ejemplo anterior, se observa la sintaxis de una función que arroja la media aritmética de dos o más números, pero es necesario recordar qué es la media aritmética. Según Paz (s. f.), “La media aritmética o promedio simple se calcula sumando los valores de interés y dividiendo entre el número de valores sumados”. Para el anterior caso, se suman 2 números y el resultado se divide entre 2, así:  Los autores, además, dicen que estos tienen 3 tipos de visibilidad: Public “Pública”, Private “Privada”, o Protected “Protegidas”, si no se especifica, se sobreentiende que es privada. |
| --- | --- | --- |

**2. Clases y objetos**

Las clases son contenedores de información basados en atributos y en métodos de construcción, los cuales pueden modificar y consultar; son suficientes para representar objetos con los cuales se procesa información, y se usan para hacer objetos que tienen un mismo comportamiento, estado e identidad.

| megáfono1 con relleno sólidoEjemplo 2:  Se tiene una clase “personas” (Mario, Margarita y Alejandro), donde el comportamiento de las personas es leer, trabajar, jugar, etc.; pueden estar en estado despierto o dormido; sus propiedades pueden ser color de ojos, género, estado civil, etc.  Su sintaxis sería así:  class Persona {  # Propiedades  # Métodos  } |
| --- |

**Objeto**

El objeto es una instancia de una clase, la cual puede crear varias instancias de la misma clase.

**Ejemplo**: Existe una sola clase “Persona”, pero muchos objetos de tipo persona pueden ser instancias de esta clase.

A continuación, observe la diferencia entre clases y objetos con una analogía:



**3. Integración de aplicaciones (XML, SOAP, WSDL y UDDI)**

Los servicios web permiten la nueva generación de aplicaciones basadas en Internet. Estos servicios admiten la comunicación de Internet de aplicación a aplicación, es decir, las aplicaciones en diferentes ubicaciones de la red se pueden integrar para funcionar como si fueran parte de un único sistema de *software* grande. Ejemplos de aplicaciones que los servicios web hacen posibles incluyen transacciones comerciales automatizadas y acceso directo (sin navegador) a dispositivos de escritorio y de mano, a las reservas, el comercio de acciones y los sistemas de seguimiento de pedidos.

Han surgido varios estándares clave que, juntos, forman la base de los servicios web: XML (Extensible Markup Language), WSDL (Web Services Definition Language), SOAP (Simple Object Access Protocol) y UDDI (Universal Description, Discovery, and Integration). Además, se ha especificado ebXML (Electronic Business XML) para facilitar la integración automatizada de procesos comerciales entre socios comerciales.

**3.1 XML “Extensible Markup Language”**

Según Bianco (2005), *XML*(Extensible Markup Language) es un lenguaje de marcas que define un conjunto de reglas para la codificación de documentos, se utiliza para estructurar información en cualquier fichero que contenga texto y es muy usado en el medio, debido a que es un estándar abierto y libre, creado por el consorcio World Wide Web, W3C.

Además, argumenta que el lenguaje XML proporciona una estructura para definir elementos, crear un formato y generar un lenguaje personalizado. Estos archivos están conformados en dos partes prolog y body:

1. **Prolog**son metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios.
2. **Body** se compone de dos partes: estructural y de contenido (presente en los textos simples).

El XML es de simple uso, por lo cual se utiliza para varios servicios web, haciendo posible una forma independiente de almacenar datos para que puedan ser compartidos por diferentes aplicaciones, logrando una óptima compatibilidad entre dispositivos.

**Estructura**

El mismo autor afirma que elPrólogo va en la primera línea de código, aquí se declara que el documento es XML y qué versión se está utilizando. Vea el siguiente ejemplo:

| megáfono1 con relleno sólidoEjemplo 3:  <?xml version="1.0"?> “Declaración normal, sirve para cualquier documento, solo especifica la versión”  <?xml versión= “1.0” encoding= “ISO-8859-1” standalone= “yes”?> “Aquí está la versión y también se agrega el tipo de codificación y el Standalone, que se refiere a que es un documento autónomo” |
| --- |

En el cuerpoes donde va la programación. Hay que tener en cuenta un elemento principal llamado raíz, dentro del cual se encuentran el resto de los elementos. Este elemento es el **“padre”** de todos los demás elementos, y de él se derivan las ramas del árbol hasta el nivel más bajo.

Hay que tener en cuenta que XML es muy diferente de HTML, por ejemplo:

**Tabla 1**

*Diferencias XML-HTML*

| XML | HTML |
| --- | --- |
| <libro>  <autor>Mario Meneses</autor>  <titulo>Guía de Aprendizaje XML</titulo>  <precio moneda=“pesos”>100.000</precio>  </libro> | <b>Mario Meneses</b>  <b><i>Guía de Aprendizaje XML</i></b>  <b>precio: 100.000 $ </b> |
| XML Utiliza etiquetas para definir el contenido y significado de esta información | HTML se centra en colocar etiquetas para representar información |

Para hacer un XML, se debe tener en cuenta la estructura de árbol, la cual es jerárquica; debe haber un elemento raíz y que de este se desprendan todos los demás, estos no se pueden superponer entre ellos, al contrario, deben estar anidados, por ejemplo, de la siguiente forma:

**Figura 1**

*Estructura*



Vea a continuación un ejemplo:



**Etiquetas o *Tags***

En el anterior ejemplo, se observa que hay elementos en los cuales se describen los datos que se contienen, así:

| megáfono1 con relleno sólidoEjemplo 4:  En el ejercicio 1, se puede observar que:  <titulo>Guía de Aprendizaje XML</titulo>   * Cuando se inicia una etiqueta, siempre se deben abrir así: <titulo> * Cuando se termina de describir un dato, siempre se debe cerrar así: </titulo> |
| --- |

**Atributos**

Un atributo sirve para proporcionar información extra sobre el elemento que lo contiene, los atributos siempre están dentro de las etiquetas de apertura:

| megáfono1 con relleno sólidoEjemplo 5:  En el ejercicio 1, se puede observar que:  <precio moneda="euros">30</precio>  <precio moneda="pesos">80.000</precio>  El atributo está entre comillas dobles, pero también puede ir entre comillas sencillas. En este caso específico, indica que la moneda de un libro es en euros mientras que el otro es en pesos  <autor>Mario Meneses</autor>  <titulo>Guía de Aprendizaje XML</titulo>  <precio moneda="euros">30</precio>  <autor>Eduardo Benavides</autor>  <titulo>XML con ejemplos</titulo>  <descriptor>lenguajes<descriptor>  <descriptor>Programación<descriptor>  <precio moneda="euros">30</precio>  <precio moneda="pesos">80.000</precio> |
| --- |

Miremos el siguiente ejercicio:

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <?xml version="1.0"?>  <Familia>  <Padres>  <Padre> nombre = “Mario”</Padre>  <Madre> nombre = “Margarita”</Madre>  </Padres>  <Hijos>  <Nombre>”Samuel” genero=”Masculino”</Nombre>  <Nombre> ”Ana” genero=“Femenino”</Nombre>  </Hijos>  <Mascotas>  <gato> nombre=”Bills”</gato>  <perro> nombre=”Vegeta”</perro>  </Mascotas>  </Familia> |
| --- | --- |

**<Padre** nombre = “Mario”**></Padre>**

**<Madre** nombre = “Margarita”**></Madre>**

**<Nombre>**”Samuel” genero=”Masculino”**</Nombre>**

**<Nombre>** ”Ana” genero=“Femenino”**</Nombre>**

**<gato> nombre=”Bills”</gato>**

**<perro> nombre=”Vegeta”</perro>**

**<gato** nombre=”Bills”**/>**

**<perro** nombre=”Vegeta**”/>**

Es posible ensayar y editar este código fuente de una manera muy sencilla.



De acuerdo con Bianco (2005), hay que recordar que:



* Para editar el código, existen muchos programas, como: Microsoft Word, bloc de notas, Notepad++ y un sin número de editores.
* Para visualizar, es mejor utilizar los navegadores Google Chrome o Mozilla Firefox, ya que otros pueden tener problemas de compatibilidad.
* XML hace distinción de la escritura en mayúsculas y minúsculas, por lo cual hay que tener cuidado en la escritura.

**3.2 SOAP “Simple Object Access Protocol”**

SOAP es un acrónimo de Protocolo Simple de Acceso a Objetos. Es un protocolo de mensajería, basado en XML, para intercambiar información entre computadoras, y es una aplicación de la especificación XML. Profundice con el siguiente video:



**3.3 WSDL “Lenguaje de Descripción de Servicios Web”**

Según Gutiérrez (2016), “es un lenguaje por medio del cual un servicio web describe, entre otras cosas, qué hace o qué funcionalidad implementa. 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”.

**Estructura de WSDL**

Este archivo tiene una estructura jerárquica, por lo tanto, la información está anidada; tiene dos tipos de descripciones:

1. **Descripciones abstractas:** se refiere a la funcionalidad del servicio.
2. **Descripciones concretas**: comunican datos específicos, como el protocolo de transmisión.

El mismo autor resalta que WSDL recurre a 6 elementos principales:



Por medio del WSDL, el cliente tiene toda la información necesaria para acceder a un servicio web, el beneficio es que todos los sistemas poseen el mismo lenguaje, lo que hace que los servicios web sean multiplataforma.

**3.4. UDDI “Universal Description, Discovery, and Integration”**

Para IBM Corporation (s. f.), UDDI (Universal Description, Discovery, and Integration) “define un modo de publicar y encontrar información sobre servicios web, incluye un esquema XML para mensajes SOAP que define un conjunto de documentos para describir información de empresas y servicios, un conjunto común de API para consultar y publicar información en los directorios y una API para duplicar entradas de directorio entre nodos UDDI iguales”.

UDDI tiene dos funciones:

* Es un protocolo basado en SOAP que define cómo se comunican los clientes con los registros UDDI.
* Es un conjunto de registros duplicados globales en particular.

**Figura 2**

*Protocolo UDDI*





**3.5 REST “Representational State Transfer”**

El protocolo REST (Representational State Transfer) es una arquitectura cliente-servidor, en la cual un servicio es visto como un recurso que es identificado a través de una dirección URL, por medio de la cual puede ser accedido o consumido. Para acceder a estos servicios web, se hace uso de mensajes en formato simple, los cuales se intercambian entre cliente y servidor.

Este define, a partir de HTTP, cuatro métodos:

**GET**

Es usado para enviar la representación de un recurso o servicio al cliente “consulta información”.

**PUT**

Este es usado para transferir el estado de un cliente al recurso “actualiza un registro”.

**DELETE**

Borra un recurso específico.

**POST**

Es utilizado para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor “crear un nuevo registro”.

El mismo autor resalta que, para hacer uso de esta comunicación, se hace a través de estos lenguajes o formatos: XML, HTML, JSON; JSON, este es el tipo de mensajes más difundido en diferentes servicios propios de redes sociales (Facebook y Twitter) y comunidades en Internet. Cada mensaje intercambiado contiene la información necesaria para el funcionamiento del servicio, de tal forma que, para cada servicio, el cliente y el servidor conocen el formato o protocolo interno de los mensajes.

**3.6 JSON “JavaScript Object Notation”**

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos, es de fácil lectura y escritura para los usuarios, fácil de analizar y generar por parte de los dispositivos, además, se basa en un subconjunto del lenguaje de programación JavaScript; es compatible con la gran mayoría de lenguajes de programación, lo que lo hace el más popular y usado en el web service. (IBM Corporation, 2020)

La cardinalidad, o también llamada multiplicidad, indica la cantidad de elementos o instancias de una entidad A que se relacionan con una instancia de una entidad B y viceversa. Esta puede ser de 4 tipos:

Uno a uno “única”

Uno a varios “múltiple”

Varios a uno “múltiple”

Varios a varios “múltiple”

A continuación, observe un ejemplo:



En el ejemplo 1, se observa una cardinalidad única, ya que es de uno a varios, mientras que en el ejemplo 2, es una cardinalidad múltiple, porque se relaciona de varios a varios.

Póngase a prueba, es hora de responder la siguiente actividad, ¡adelante!



**4. Creación de servicios**

Sayago et al. (2019) indican que es una tecnología que establece unos estándares y protocolos que se utilizan para intercambiar datos entre distintas aplicaciones, plataformas y con diferentes o iguales lenguaje de programación. Por lo general, se utilizan estándares abiertos, lo que comúnmente se llama interoperabilidad; las organizaciones [OASIS](https://es.wikipedia.org/wiki/OASIS_(organizaci%C3%B3n)) y [W3C](https://es.wikipedia.org/wiki/World_Wide_Web_Consortium) son los comités responsables de la arquitectura y la reglamentación de los servicios web.

Los mismos autores hacen notar que su función es permitir que los dispositivos interactúen entre sí y presenten información dinámica al usuario, con base en una arquitectura de referencia estándar, que hace posible su combinación para realizar integraciones entre las mismas aplicaciones. Para la creación de servicios, se debe seguir el siguiente proceso:



**5. Conexiones a SQL**

Para iniciar con ese tema, es preciso recordar que el SQL (Structured Query Language) es un lenguaje de consulta estructurado, definido como la base de datos relacionales con el lenguaje de alto nivel estándar.

De acuerdo con García (2003), el SQL agrupa tres tipos de sentencias con objetivos particulares, en los siguientes lenguajes:

* (DDL) Lenguaje de Definición de Datos
* (DML) Lenguaje de Manipulación de Datos
* (DCL) Lenguaje de Control de Datos

El mismo autor plantea que el grupo de sentencias de SQL soportan la definición y declaración de los objetos de la base de datos. Objetos tales como:

* **DATABASE:** base de datos misma.
* **TABLE:** tablas.
* **VIEW:** vistas.
* **INDEX:** índices.
* **PROCEDURE**: los procedimientos almacenados.
* **TRIGGER:** los disparadores.
* **DEFAULT:** valores por defecto CREATE, ALTER y DROP.

Para el ejemplo que se realizará más adelante, en el cual se unirán todos los temas, se utilizará phpMyAdmin, que es una herramienta escrita en PHP cuya función es administrar MySQL a través de páginas web, utilizando cualquier navegador.

**6. Servicios en PHP**

PHP es uno de los lenguajes de código abierto más utilizados y adecuados para el desarrollo de aplicaciones y servicios web, este puede ser introducido en HTML. Por el hecho de ser un lenguaje gratuito, millones de páginas y portales web están creados con PHP. Su función es muy sencilla: el usuario utiliza aplicaciones o navegadores para realizar una solicitud a un servidor, el servidor recibe la petición, organiza la información que le solicitan y responde enviando la consulta por medio de una página web o aplicación “normal”, pero su creación es “dinámica”.

**Tabla 2**

*PHP*

| Páginas estáticas | Petición | Respuesta |
| --- | --- | --- |
| Páginas dinámicas | Petición | Procesa, prepara y da respuesta |

El proceso de los servicios PHP está estructurado de la siguiente manera:



Ejercicio práctico creación de servicios web:

|  | En el siguiente ejercicio, se observarán en detalle los temas que se han abordado en cuanto a servicios web. Se trabajará desde el punto de vista del creador de servicios y el cliente o consumidor de servicios:   * XAMPP * Editor de código fuente de licencia libre como = Notepad++ o NetBeans * Navegador web = Chrome o Internet Explorer |
| --- | --- |

**7. Aplicación SOAP UI**

Según la organización SoapUI (s. f.), la aplicación SOAP UI esta es una herramienta muy versátil, que sirve para probar, simular y obtener el código de aplicaciones *web service* y la transferencia de estado representacional. Esta herramienta está generada en código java, funciona sobre SOAP, JMS, JDBC, API de REST, GraphQL y muchos servicios más, además de ser muy útil para que los futuros desarrolladores aprendan a crear API.

|  | Para descargar este programa de forma gratuita, haga clic aquí. |
| --- | --- |

**7.1 Instalación y configuración**

Para la instalación del programa, se deben seguir los siguientes pasos:



**7.2 Uso de la herramienta SOAP UI**

Para probar un servicio web existente, se utilizan recursos y directorios *web service* que se pueden acceder de manera libre, o diferentes páginas que proveen servicios, como almacenes, bibliotecas, supermercados, etc.

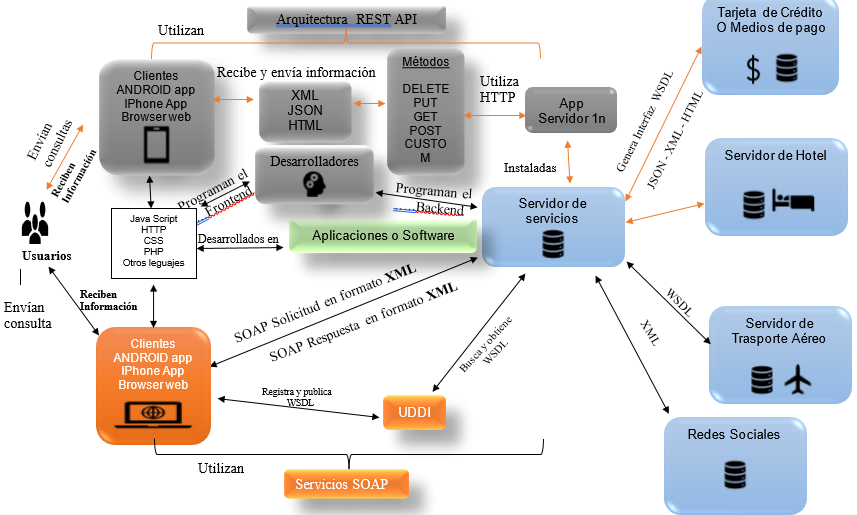
Para esto, es necesario saber qué tipo de proyecto se utilizará en SOAP UI, si es REST o SOAP.

| Internet con relleno sólido | Para este caso en particular, se utilizará una página:  <https://regres.in/> |
| --- | --- |
| Diseño web con relleno sólido | Se realizará la prueba con el ejercicio que se desarrolló:  <http://localhost/trabajo/cliente/index.php> |
| Vlog con relleno sólido | :  Funcionamiento de herramienta Soapui  Anexo5\_Soapui pruebas de funcionamiento |

Ha llegado al final de este componente formativo, pero, antes de concluir, realice esta actividad:



**Síntesis**



1. **Actividades didácticas (opcionales si son sugeridas)**

| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la Actividad | Conceptos básicos de servicios web con PHP |
| Objetivo de la actividad | Afianzar algunos conceptos básicos de servicios web con PHP |
| Tipo de actividad sugerida | Arrastrar y soltar el término sobre la definición que corresponde. |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo documento en Word llamado Actividad didáctica 1 |

| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la Actividad | Conocimientos de servicios web |
| Objetivo de la actividad | Afianzar algunos de los conceptos de servicios web |
| Tipo de actividad sugerida | Indicar si cada uno de los enunciados es verdadero o falso. |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo documento en Word llamado Actividad didáctica 2 |

1. **Material complementario**

| **Tema** | **Referencia APA del Material** | **Tipo de material**  **(Video, capítulo de libro, artículo, otro)** | **Enlace del Recurso o**  **Archivo del documento o material** |
| --- | --- | --- | --- |
| Integración de aplicaciones (XML, SOAP, WSDL y UDDI) | IBM (s. IBM Corporation. (s. f.). *UDDI (Universal Description, Discovery, and Integration).* <https://www.ibm.com/docs/es/rsas/7.5.0?topic=standards-universal-description-discovery-integration-uddi> | Página web | <https://www.ibm.com/docs/es/rsas/7.5.0?topic=standards-universal-description-discovery-integration-uddi> |
| Servicios en PHP | Meneses, M. (2021a). *Creación archivos PHP en el localhost* [Video]. YouTube. <https://youtu.be/9MXAQGGxCN4> | Video | <https://youtu.be/9MXAQGGxCN4> |
| Servicios en PHP | Meneses, M. (2021b). *Web Services programación Front End Consumo de servicios web* [Video]. YouTube. <https://youtu.be/MV3dSZx5iTo> | Video | <https://youtu.be/MV3dSZx5iTo> |
| Aplicación SOAP UI | SoapUI. (s. f.). *Accelerating API Quality Through Testing*. <https://www.soapui.org/> | Página web | <https://www.soapui.org/> |

1. **Glosario**

| TÉRMINO | SIGNIFICADO |
| --- | --- |
| **API** | La interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro *software* como una capa de abstracción. |
| **BackEnd** | Es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos, como, por ejemplo, la comunicación con el servidor. |
| **FrontEnd** | El desarrollo web FrontEnd consiste en la conversión de datos en una interfaz gráfica, para que el usuario pueda ver e interactuar con la información de forma digital, usando HTML, CSS y JavaScript. |
| **REST** | La Transferencia de Estado Representacional, o REST, es un estilo de arquitectura *software* para sistemas hipermedia distribuidos, como la World Wide Web. |

1. **Referencias bibliográficas**

Álvarez, J. (2017). *Entorno de programación intencional basado en XML*. Universidad Politécnica de Madrid. <https://oa.upm.es/49793/1/PFC_JOSE_ANTONIO_ALVAREZ_PEREZ.pdf>

Bianco, P. (2005). *Desarrollo de Aplicaciones Basadas en XML Web Service para Dispositivos Móviles con Microsoft.* *NET Compact Framework*. Universidad de Belgrano. <http://repositorio.ub.edu.ar/handle/123456789/343>

Chanchí, G., Campo, W., Amaya, J. y Arciniegas, J. (2011). Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON. *Cuadernos de Informática, 6*(1), p. 233-240. <http://seer.ufrgs.br/cadernosdeinformatica/article/view/v6n1p233-240>

Eslava, V. (2013). *El nuevo PHP. Conceptos avanzados*. Bubok.

García, A. (2003). *Manual práctico de SQL*. <https://www.lawebdelprogramador.com/cursos/archivos/ManualPracticoSQL.pdf>

Gonzáles, S. y Pelissier, C. (2002). *Programación con PHP*. Universidad Técnica Federico Santa María. <http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/pelissier/informe.pdf>

Gutiérrez, A. (2016). *Elaboración de un servicio web para el registro de operaciones entre clientes:(infraestructura de fibra óptica NEBA de Telefónica)*. Universidad Carlos III de Madrid. <https://e-archivo.uc3m.es/bitstream/handle/10016/22335/PFC_Alvar_Guti%C3%A9rrez_Rom%C3%A1n.pdf>

IBM Corporation. (2015). *IBM integration Bus 10.0.0.* [https://www.ibm.com/docs/es/integration-bus/10.0?topic=ssmkhh-10-0-0-com-ibm-etools-mft-doc-bi12017—htm](https://www.ibm.com/docs/es/integration-bus/10.0?topic=ssmkhh-10-0-0-com-ibm-etools-mft-doc-bi12017%E2%80%94htm)

IBM Corporation. (2020). *Formato Json (JavaScript Object Notation).* <https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format>

IBM Corporation. (s. f.). *UDDI (Universal Description, Discovery, and Integration).* <https://www.ibm.com/docs/es/rsas/7.5.0?topic=standards-universal-description-discovery-integration-uddi>

Ortiz, A., Otón, S. y Barchino, R. (2005*). Learning Objects universal publishing and location Architecture using Web Services*. Universidad de Alcalá. <https://www.researchgate.net/publication/267217723_Learning_Objects_universal_publishing_and_location_Architecture_using_Web_Services>

Paz, K. (s. f.). *Media aritmética simple*. Universidad Rafael Landívar. <https://fgsalazar.net/LANDIVAR/ING-PRIMERO/boletin07/URL_07_BAS01.pdf>

Sayago, J., Flores, E. y Recalde, A. (2019). Análisis comparativo entre los estándares orientados a servicios web SOAP, REST y GRAPHQL. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS), 9*(2), p. 10-22. <http://dx.doi.org/10.5281/zenodo.3592004>

SoapUI. (s. f.). *Acelerando la calidad de la API a través de pruebas.* <https://www.soapui.org/>

Villate, J. (2001). *Introducción al XML*. Universidad de Oporto. <https://docplayer.es/1640770-Introduccion-al-xml-jaime-e-villate-universidad-de-oporto-villate-fe-up-pt-5-de-mayo-de-2001.html>

1. **Control del documento**

|  | Nombre | Cargo | Dependencia | Fecha |
| --- | --- | --- | --- | --- |
| Autor(es) | Mario Fernando Meneses Calvache | Experto Temático | Regional Cauca, Centro de teleinformática y producción industrial | Septiembre 2021 |
| Jonathan Guerrero Astaiza | Instructor | Regional Cauca, Centro de teleinformática y producción industrial | Septiembre 2021 |
| Zulema León | Instructor | Regional Cauca, Centro de teleinformática y producción industrial | Septiembre 2021 |
| Carlos Hernán Muñoz | Experto Temático | Regional Cauca, Centro de teleinformática y producción industrial | Septiembre 2021 |
| Paola Moya Peralta | Diseñadora instruccional | Regional Distrito Capital - Centro para la Industria de la Comunicación Gráfica | Octubre 2021 |
| Ana Catalina Córdoba Sus | Revisora metodológica y pedagógica | Regional Distrito Capital – Centro para la Industria de la Comunicación Gráfica. | Noviembre 2021 |
| Rafael Neftalí Lizcano Reyes | Asesor pedagógico | Regional Santander - Centro Industrial del Diseño y la Manufactura. | Noviembre 2021 |
| Darío González | Corrección de estilo | Regional Tolima – Centro Agropecuario La Granja | Diciembre 2021 |

1. **Control de cambios**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  | Nombre | Cargo | Dependencia | Fecha | Razón del cambio |
| --- | --- | --- | --- | --- | --- |
| Autor(es) |  |  |  |  |  |