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.

Julio 2024

Grupo de Ejecución de la Formación Virtual

Tabla de contenido

Introducción

En el mundo de la programación y desarrollo de software, la organización y estructuración del código son fundamentales para asegurar la eficiencia y facilidad de mantenimiento de las aplicaciones. Este documento aborda conceptos esenciales como métodos o funciones, clases y objetos, y la integración de aplicaciones mediante estándares como XML, SOAP, WSDL y UDDI. Además, se explora el proceso de creación de servicios, destacando su desarrollo, ejecución y consumo. Estos temas son cruciales para los desarrolladores que buscan construir sistemas robustos y escalables, capaces de interactuar con diversas aplicaciones y plataformas. A través de una comprensión clara y concisa de estos elementos, se sientan las bases para la creación de soluciones tecnológicas que respondan a las necesidades actuales del mercado.

La correcta utilización de métodos o funciones permite segmentar el trabajo en tareas más pequeñas y manejables, facilitando su reutilización y mantenimiento. Por otro lado, la definición clara de clases y objetos es fundamental para la organización del código, proporcionando una estructura lógica que mejora la eficiencia del desarrollo y la claridad del programa. Estos conceptos no solo optimizan el proceso de programación, sino que también permiten una mejor colaboración entre equipos de desarrollo, ya que el código se vuelve más legible y comprensible.

La integración de aplicaciones mediante XML, SOAP, WSDL y UDDI es un componente vital en la creación de servicios web modernos. Estos estándares permiten que diferentes aplicaciones se comuniquen e interactúen entre sí, independientemente de las plataformas o lenguajes de programación utilizados. Al dominar estos procesos, los desarrolladores pueden crear servicios que sean accesibles y utilizables por una

amplia gama de sistemas, asegurando la interoperabilidad y ampliando el alcance y la funcionalidad de las aplicaciones. Esta integración es esencial para construir soluciones tecnológicas que sean tanto versátiles como adaptables a las cambiantes demandas del mercado.

Le deseamos muchos éxitos en este proceso de aprendizaje.

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. Con el nombre de la función, es posible invocarla las veces que se necesite, permitiendo utilizar esta función repetidamente según sea necesario.

Ejemplo:

La función se define con la siguiente sintaxis:

```
<?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),"
"; ?> //aquí ingresamos otros dos números 300 y 500

Nota: el símbolo "//" quiere decir que no hace parte del código, se utiliza para hacer comentarios sobre el código, por lo cual se pone en color azul.

La función devuelve como resultados los números 8 y 400.

En el ejemplo anterior, se describe la sintaxis de una función que calcula 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 caso anterior, se suman 2 números y el resultado se divide entre 2, así:

$$\frac{\$a + \$b}{2}$$

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.

Ejemplo:

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, revise la diferencia entre clases y objetos con una analogía:

- **Una clase**

  Es como el plano de un apartamento, define la forma del apartamento sobre el papel, con las relaciones entre las diferentes partes claramente definidas y planeadas, a pesar de que el apartamento no exista.

- **Un objeto**

  Es el apartamento real construido de acuerdo a ese modelo; es decir, un objeto es una instancia de la clase.

Ejemplo:

En el siguiente ejemplo, se observará el código de un objeto.

```
<?php
 class Persona // Se define la clase Persona
 {
  private $nombre; // Declaración del campo. Este debe ser de carácter privado.

  // Aquí adelante se utiliza un constructor, que es un método que lleva el nombre de la clase y que será el responsable de construir el objeto.
  function Persona($nom)
  {
   $this->nombre = $nom; // this hace referencia a la instancia actual, y se le asigna el valor al constructor->
```

```
    }

    function setNombre($n) { // Métodos accesorios
      $this->nombre = $n; // Métodos de escritura
    }

    function getNombre() { // Métodos de lectura
      return $this->nombre;
    }

    public function Saluda($saludo) { // Métodos se define como público, solo es un mensaje
      echo $saludo."<br>";
    }
  }

  $objPersona = new Persona("Mario Meneses"); // Se define el nombre del Objeto
  echo "<p>El objeto <span style='color:red; font-weight:bold'>objPersona</span> se llama " .
```

$objPersona->getNombre()."</p>"; // se imprime el texto “el objeto se llama con estilos de color”

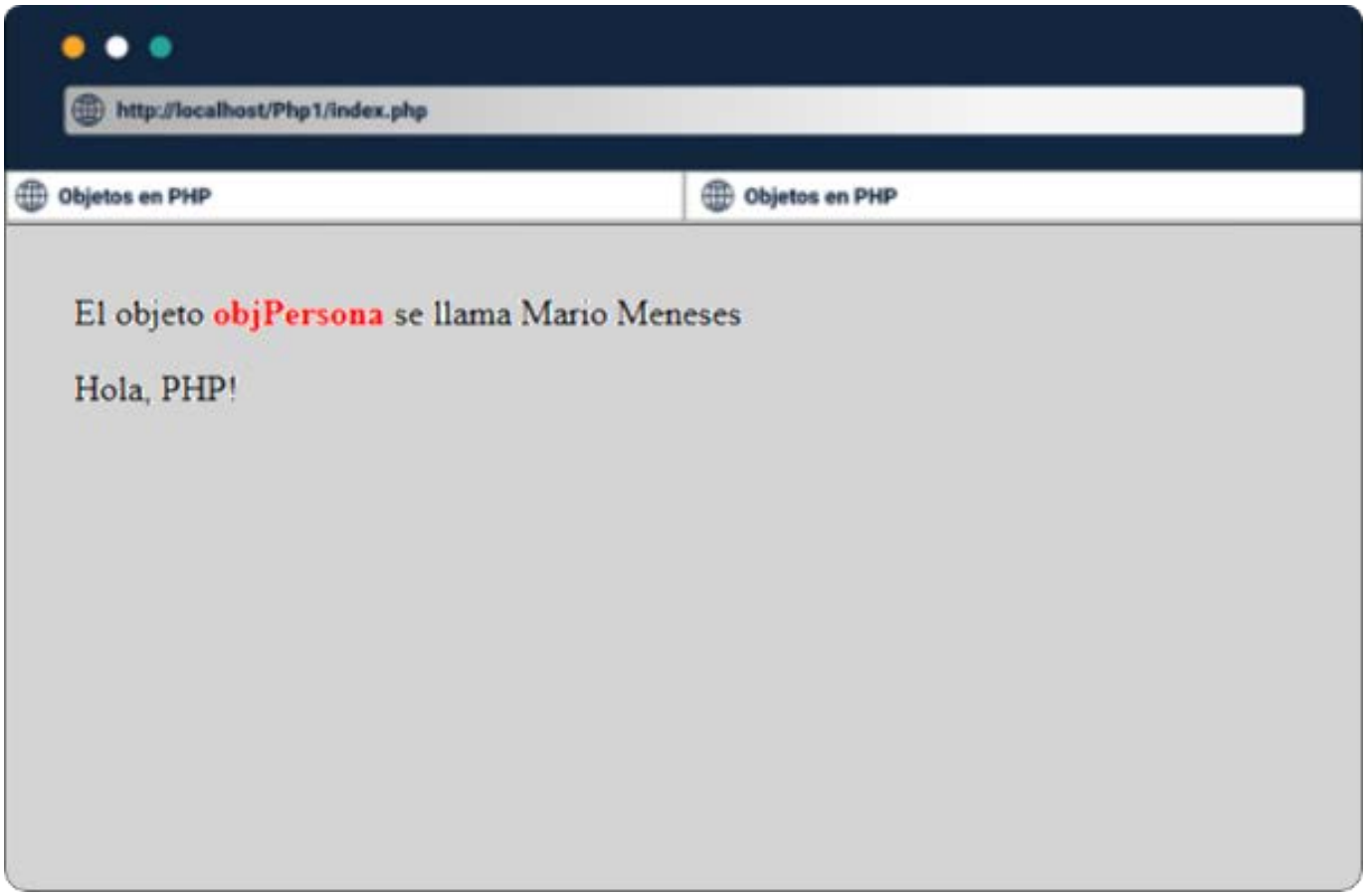
$objPersona->Saluda("Hola, PHP!"); // se imprime el nombre escrito

?>

Resultado

Si se ejecutara el código, daría este resultado.

Figura 1. Resultado del ejemplo



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

"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". Bianco (2005)

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

- **PROLOG**

  Son metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios.

- **BODY**

  Se compone de dos partes: estructural y de contenido (presente en los textos simples).

El XML es de uso sencillo, por lo cual se utiliza en 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 prólogo 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:

Ejemplo:

<?xml version="1.0"?>

Declaración normal, sirve para cualquier documento, solo especifica la versión.

<?xml version="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 cuerpo es 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.

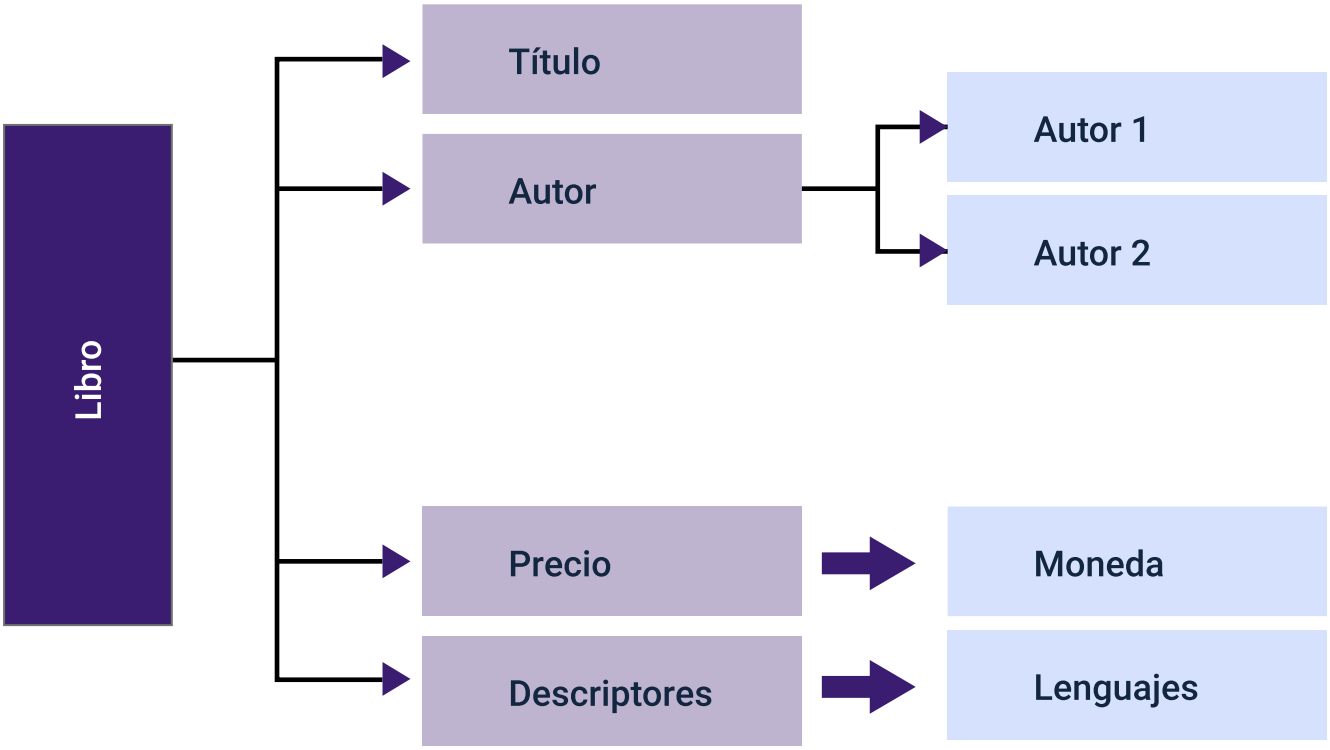
Hay que tener en cuenta que XML es muy diferente de HTML, por ejemplo:

Tabla 1. Diferencias XML-HTML

XML	Título de col2
<libro> <autor>Mario Meneses</autor> <titulo>Guía de Aprendizaje XML </titulo> <precio moneda="pesos">100.000 </precio> </libro>	Mario Meneses Guía de Aprendizaje XML precio: 100.000 $
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 del cual se desprenden todos los demás. Estos no se pueden superponer entre ellos; al contrario, deben estar anidados. Por ejemplo, de la siguiente forma:

Figura 2. Estructuras



Etiquetas o Tags

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

Ejemplo:

En el ejercicio, se puede identificar 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:

Ejemplo:

En el ejercicio, 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>

Revisemos el siguiente ejercicio:

```
<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>
```

Es posible probar y editar este código fuente de una manera muy sencilla:

- **Paso 1:**

  Abrir el programa Notepad++

- **Paso 2:**

  Pegar o escribir el código.

- **Paso 3:**

  Dar clic en guardar como, seleccionar una ubicación y, en el nombre, escribir un nombre+.xml, así: Familia.xml

- **Paso 4:**

  Guardar los cambios.

- **Paso 5:**

  Ahora, al archivo que guardó, se le da clic derecho y se abre con el navegador Google Chrome, y debe aparecer en el navegador así:

```
<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>
```

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:

Video 1. SOAP Simple Object Access Protocol

Enlace de reproducción del video

Síntesis del video: SOAP Simple Object Access Protocol
SOAP, sus siglas hacen referencia a Simple Object Access Protocol, es un protocolo estándar que define cómo dos objetos en diferentes procesos, deben comunicarse por medio de intercambio de datos XML. Las tiendas en línea, los buscadores y las empresas que ofrecen productos en línea aplican a este protocolo, el cual se utiliza desde los años 90, es la conexión entre un cliente navegador de internet y los servicios de un servidor. La función básica de SOAP, es comunicar aplicaciones realizando la llamada a procedimiento remoto RPC e implementando un esquema requerimiento respuesta

equivalente a una arquitectura cliente servidor en la cual el cliente inicia el proceso y el servidor responde al requerimiento, el cliente envía un mensaje request vía http que es generado por la aplicación servidora y una respuesta que es enviada a la aplicación cliente vía http.

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:

- **Descripciones abstractas**

  Se refiere a la funcionalidad del servicio.

- **Descripciones concretas**

  Comunican datos específicos, como el protocolo de transmisión.

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

- **Paso 1:**

  Types. Tipos de datos.

- **Paso 2:**

  Message. Descripción de los datos a transferir.

- **Paso 3:**

  Interface. Operaciones abstractas que describen la comunicación entre el servidor y el cliente (aún se llamaba portType en una versión anterior del estándar).

- **Paso 4:**

  Binding. Información sobre el protocolo de transporte utilizado.

- **Paso 5:**

  Endpoint. Información sobre la interfaz de comunicación, generalmente, en forma de un URI (aún se denominaba port en una versión anterior del estándar).

- **Paso 6:**

  Service. Puntos de acceso del servicio web.

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.

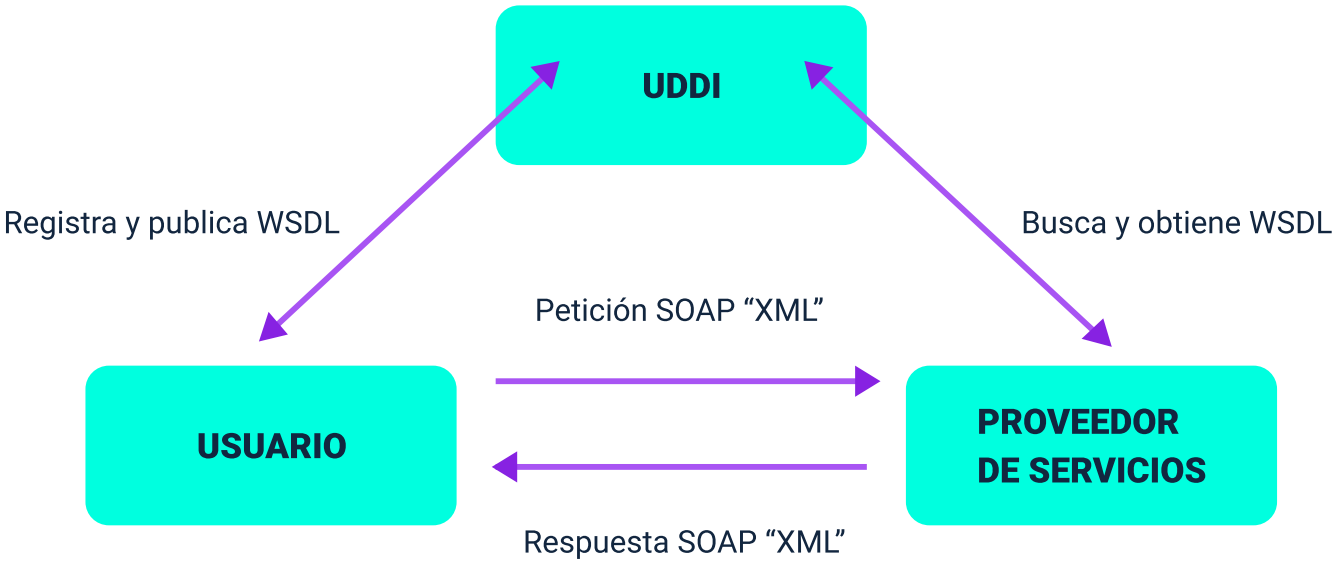
3.4. UDDI Universal Description, Discovery, and Integration

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 3. 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".

Para hacer uso de esta comunicación, se emplean los siguientes lenguajes o formatos: XML, HTML y JSON. JSON 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 y 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 los web services. (IBM Corporation, 2020).

La cardinalidad, 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 cuatro tipos:

- **Uno a uno (única)**

  Una instancia de la entidad A se relaciona con una sola instancia de la entidad B.
- **Uno a varios (múltiple)**

  Una instancia de la entidad A se relaciona con múltiples instancias de la entidad B.
- **Varios a uno (múltiple)**

  Múltiples instancias de la entidad A se relacionan con una sola instancia de la entidad B.
- **Varios a varios (múltiple)**

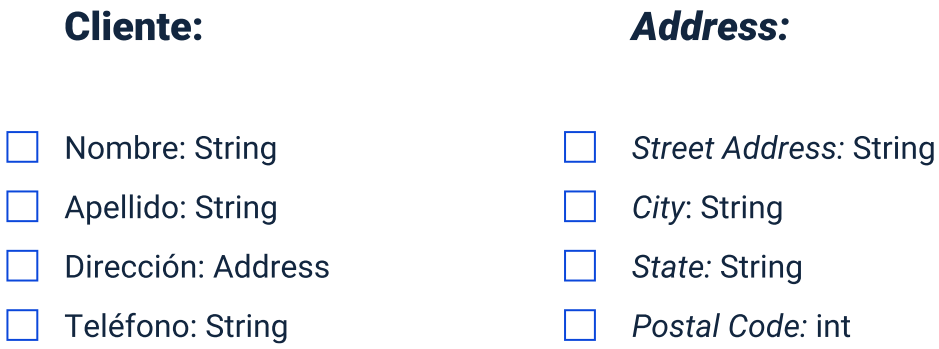
  Múltiples instancias de la entidad A se relacionan con múltiples instancias de la entidad B.

A continuación, se presentan los siguientes ejemplos:

Ejemplo 1

Se va a crear un objeto con el nombre negocio con cardinalidad única:

Figura 4. Objeto con cardinalidad única



Se asignan los siguientes valores para las propiedades del objeto negocio:

a) **Objeto:**

- Cliente

b) **Propiedad:**

- nombre
- apellido
- streetAddress
- city
- state
- codigopostal
- telefonos[0]
- telefonos[1]

c) **Valor:**

- Alejandro
- Meneses
- Calle 5 con carrera 26
- Santa Marta

- Magdalena
- 760004
- 50158182332
- 61745896258

El formato Json quedaría de la siguiente forma:

{

"nombre": "Alejandro",

"apellido": "Meneses",

"address": {

"streetAddress": "Calle 5 con carrera 26",

"city": "Santa Marta",

"state": "Magdalena",

"postalCode": 760004

},

"telefonos": [

"50158182332",

"61745896258"

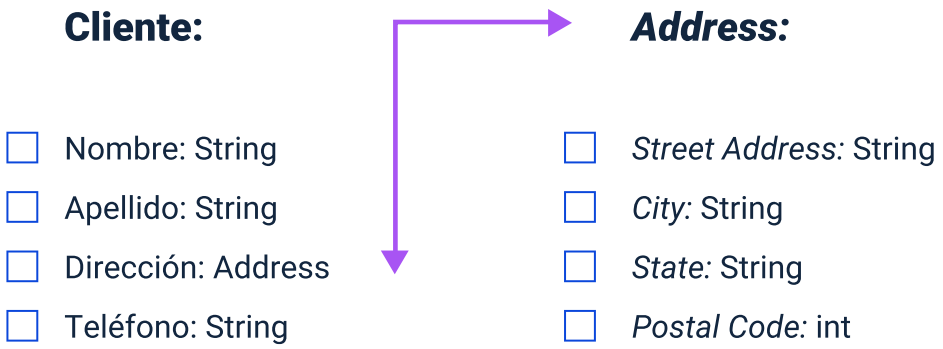
]

}

Ejemplo 2

Se va a crear un objeto con el nombre negocio con cardinalidad múltiple:

Figura 5. Objeto con cardinalidad múltiple



También, se asignan los siguientes valores para las propiedades del objeto negocio:

a) **Objeto:**

- Address []
- Address []

b) **Propiedad:**

- nombre
- apellido
- streetAddress
- city
- state
- postalCode
- telefonos[0]
- telefonos[1]

c) **Valor:**

- Alejandro
- Meneses
- Calle 5 con carrera 26
- Santa Marta
- Magdalena
- 760004
- Calle 18 No. 25-10
- Popayán
- Cauca
- 761115
- 50158182332
- 61745896258

El formato Json quedaría de la siguiente forma:

{

"nombre": "Alejandro",

"apellido": "Meneses",

"address": [

{

"streetAddress": "Calle 5 con carrera 26",

"city": "Santa Marta",

"state": "Magdalena",

"codigopostal": 760004

},

{

"streetAddress": "Calle 18 No 25-10",

"city": "Popayán",

"state": "Cauca",

"postalCode": 760004

}

],

"phoneNumbers": [

"50158182332",

"61745896258"

]

}

4. Creación de servicios

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 lenguajes de programación. Por lo general, se utilizan estándares abiertos, lo que comúnmente se llama interoperabilidad; las organizaciones OASIS y W3C son los comités responsables de la arquitectura y la reglamentación de los servicios web. Sayago et al. (2019)

La función es permitir que los dispositivos interactúen entre sí y presenten información dinámica al usuario, basándose en una arquitectura de referencia estándar, que hace posible su combinación para realizar integraciones entre las mismas aplicaciones.

Creación de servicios

Para la creación de servicios, se debe seguir el siguiente proceso:

- **Desarrollo de servicios**

  Se realiza en la parte de servicios o en la nube, y este tipo de desarrollo también se conoce como backend. Esta parte de la programación no es visible para el usuario y no se ejecuta en los dispositivos o navegadores del cliente. Aquí es donde se provee el servicio para que los clientes lo consuman (Recalde, 2019).

- **Ejecución de servicios**

  Se realiza en la parte de servicios o en la nube, y este tipo de desarrollo también se conoce como backend. Esta parte de la programación no es visible para el usuario y no se ejecuta en los dispositivos o navegadores del

cliente. Aquí es donde se provee el servicio para que los clientes lo consuman (Recalde, 2019).

- **Consumo de servicios**

  Una vez el servicio es creado, es el turno de los usuarios para consumirlo. Por ello, se desarrollan o programan aplicaciones del lado del frontend para consumir dichos servicios; esto consiste en la conversión de datos en una interfaz gráfica, para que el usuario pueda interactuar con la información.

5. Conexiones a SQL

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

De acuerdo con García (2003), 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 grupo de sentencias de SQL soporta la definición y declaración de los objetos de la base de datos, tales como:

DATABASE: Base de datos misma.

PROCEDURE: Los procedimientos almacenados.

TRIGGER: Los disparadores.

DEFAULT: Valores por defecto.

CREATE, ALTER y DROP INDEX: Índices.

VIEW: Vistas.

TABLE: Tablas.

Grupo de sentencias de SQL

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 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 solicitada 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:

- **Creación de servicios en PHP**

  Los servicios web son una integración de aplicaciones web, básicamente se trata de un servidor que expone un segmento de su funcionalidad para que sus clientes (dispositivos, aplicaciones, computadores u otros servidores) utilicen sus servicios. En su gran mayoría, los servicios se constituyen en JavaScript, que también es un lenguaje de programación, pero igualmente funciona de buena manera con PHP.

- **Implementación de servicios en PHP**

  García (2003) describe que, para que un cliente tenga acceso a la información de un servicio, lo implemente y active el método en el

servidor, hay un sistema de catálogo que, de manera predeterminada, efectúa todos los servicios web y sus métodos. De esta forma, con las credenciales adecuadas, un cliente tiene acceso a todas las funciones de los servicios web.

Al momento de implementar un servicio web, se actualiza el WSDL, archivo de Lenguaje de Descripción de Servicios Web. Hay que recordar que cada servicio web tiene su propio WSDL, el cual contiene información sobre los servicios implementados en ese momento y también las firmas de métodos.

- **Consumo de métodos**

  El consumo de un servicio, en este caso en RESTFul, es tan sencillo como enviar un pedido en HTTP y procesar su respuesta.

  Ejemplo:

  $res = file_get_contents("https://www.sena.edu.co/es-co/formacion/Paginas/Estudie-en-el-SENA.aspx");

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

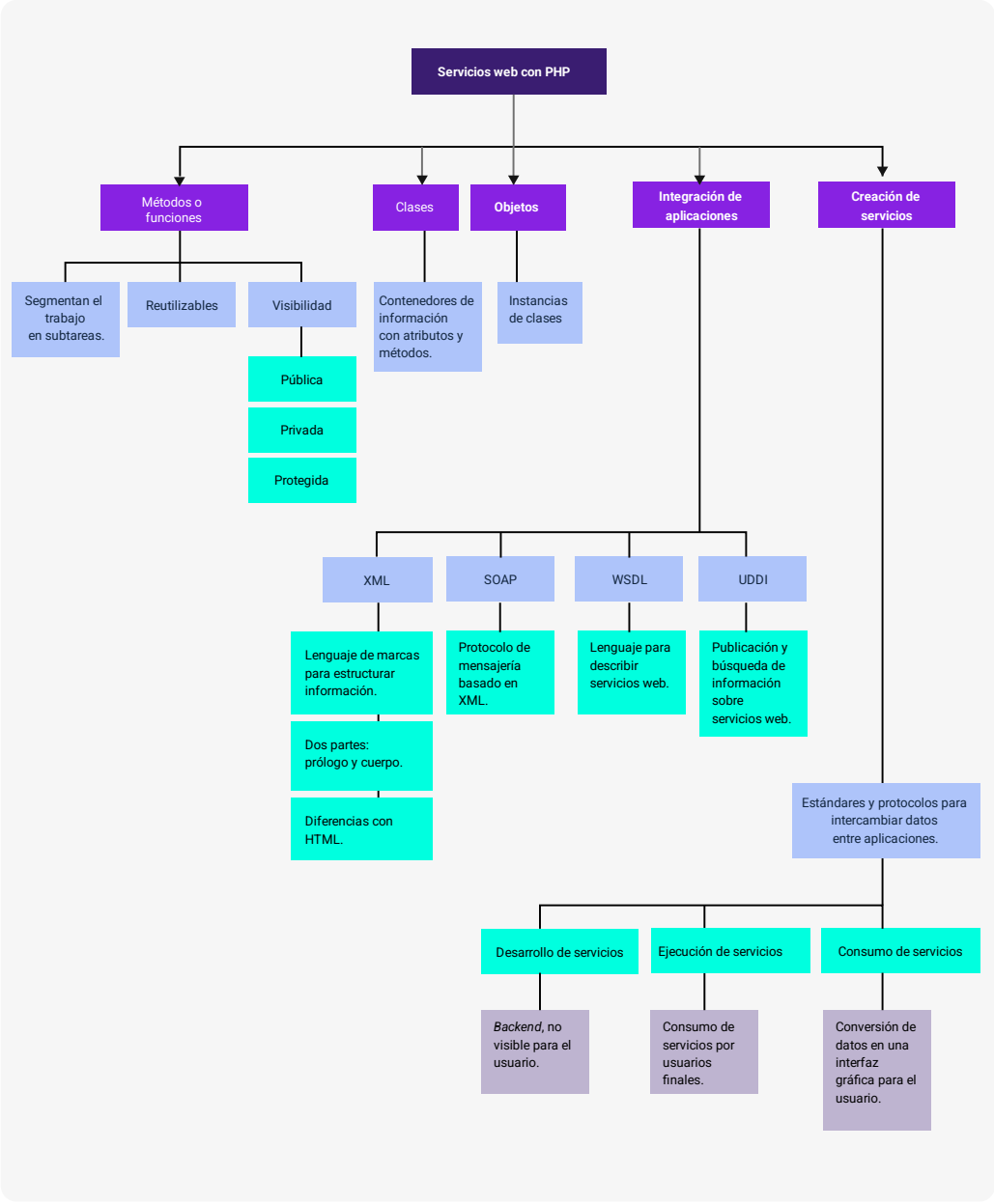
Según la organización SoapUI (s. f.), la aplicación SOAP UI es una herramienta muy versátil que sirve para probar, simular y obtener el código de aplicaciones web service y la transferencia de estado representacional. Esta herramienta está generada en código Java, funciona sobre SOAP, JMS, JDBC, API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

Uso de la herramienta SOAP UI

Para probar un servicio web existente, se utilizan recursos y directorios de web service que se pueden acceder de manera libre, o diferentes páginas que proveen servicios, como almacenes, bibliotecas, supermercados, entre otros. Para esto, es necesario saber qué tipo de proyecto se utilizará en SOAP UI, ya sea REST o SOAP.

Síntesis

A continuación, se muestra un mapa conceptual con los elementos más importantes desarrollados en este componente.



Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
Integración de aplicaciones (XML, SOAP, WSDL y UDDI)	IBM (s. IBM Corporation. (s. f.). UDDI (Universal Description, Discovery, and Integration).	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.	Video	https://youtu.be/9MXAQGGxCN4
Servicios en PHP	Meneses, M. (2021b). Web Services programación Front End Consumo de servicios web [Video]. YouTube.	Video	https://youtu.be/MV3dSZx5iTo
Aplicación SOAP UI	SoapUI. (s. f.). Accelerating API Quality Through Testing.	Página web	https://www.soapui.org/

Glosario

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.

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.

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.

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

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://villate.org/publications/Villate_2001_XML.pdf

Créditos

Nombre	Cargo	Centro de Formación y Regional
Milady Tatiana Villamil Castellanos	Responsable del Ecosistema	Dirección General
Olga Constanza Bermúdez Jaimes	Responsable de Línea de Producción	Centro de Servicios de Salud - Regional Antioquia
Jonathan Guerrero Astaiza	Experta Temática	Centro de Teleinformática y Producción Industrial - Regional Cauca
Mario Fernando Meneses Calvache	Experto Temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
Paola Alexandra Moya Peralta	Evaluadora Instruccional	Centro de Servicios de Salud - Regional Antioquia
Carlos Julián Ramírez Benítez	Diseñador de Contenidos Digitales	Centro de Servicios de Salud - Regional Antioquia
Edwin Sneider Velandia Suarez	Desarrollador Fullstack	Centro de Servicios de Salud - Regional Antioquia
Edgar Mauricio Cortés García	Actividad Didáctica	Centro de Servicios de Salud - Regional Antioquia
Daniela Muñoz Bedoya	Animador y Productor Multimedia	Centro de Servicios de Salud - Regional Antioquia
Luis Gabriel Urueta Álvarez	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Margarita Marcela Medrano Gómez	Evaluador para Contenidos Inclusivos y Accesibles	Centro de Servicios de Salud - Regional Antioquia
Daniel Ricardo Mutis Gómez	Evaluador para Contenidos Inclusivos y Accesibles	Centro de Servicios de Salud - Regional Antioquia