GC-F-004v.08

Análisis y desarrollo de software

FASE 3 Ejecución – GA7-AA5

Noviembre 19 de 2025

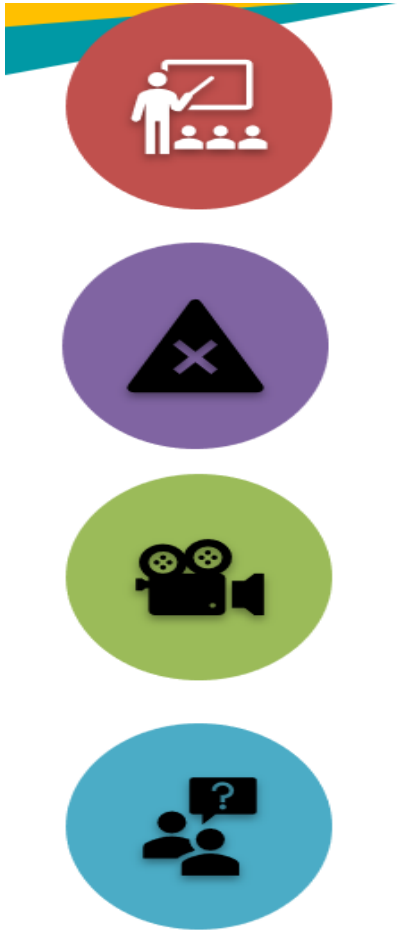
Agenda del día

1. Saludo de Bienvenida.
2. Recomendaciones Generales
3. Evidencias a desarrollar

   ✓**Evidencia de producto: GA7-220501096-AA5-EV02 API .**
   ✓**Evidencia de desempeño: GA7-220501096-AA5-EV03: Diseño y desarrollo de servicios web - proyecto .**

4. Resolución de Dudas e Inquietudes
5. Despedida Equipo de Instructores

Recomendaciones generales



1. Estudiar el material de la semana antes de la videoconferencia.
2. Evita distractores durante la sesión.
3. Toma nota de las dudas y preguntas que surjan durante la sesión y ten presente las que te hayan surgido durante el estudio del material.
4. Al final de la sesión, se asignarán 10 minutos para resolver todas las inquietudes que se hayan presentado.

Actividad de aprendizaje GA7-220501096-AA5 - Crear servicios web

Esta actividad se centra en el aprendizaje para la construcción de API´s las cuales son necesarias para el cumplimiento de necesidades en la codificación del software del proyecto.

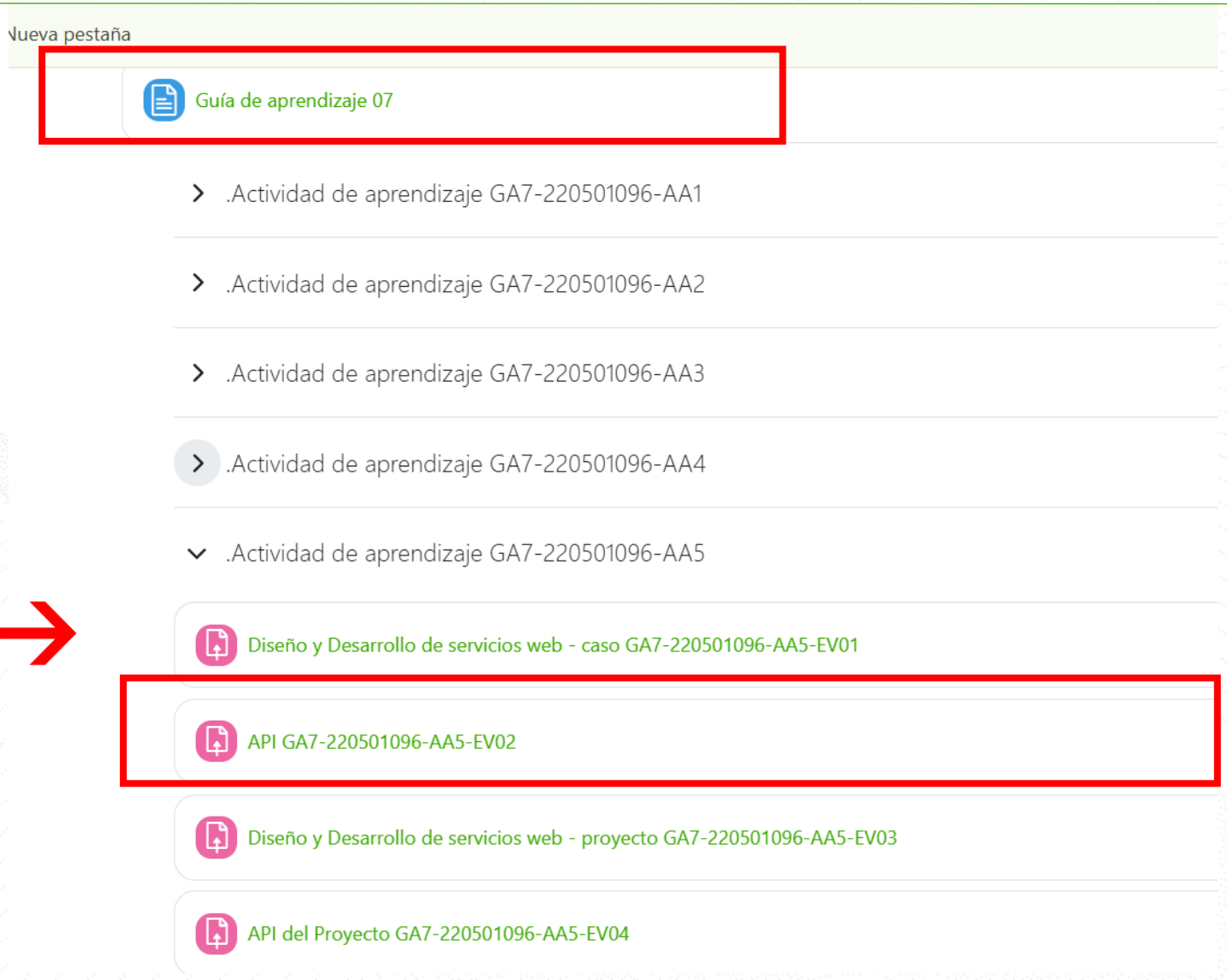
Duración: 144 horas

Materiales de formación a consultar: para el desarrollo de esta actividad es importante la lectura y análisis del componente formativo: "Construcción de API” y “Servicios web con PHP”.

Evidencias: a continuación, se describen las acciones y las correspondientes evidencias que conforman la actividad de aprendizaje**:**

Evidencia de producto: GA7-220501096-AA5-EV02 API .

Evidencia a desarrollar



Actividad de Proyecto 7

Material de formación

- Aplicación del paradigma orientado a objetos
- Integración continua
- Construcción de aplicaciones con JAVA
- Frameworks para construcción de aplicaciones con JAVA
- Desarrollo de frontend con React JS
- Construcción de API
- Servicios web con PHP
- Construcción aplicación web

Evidencia de producto: GA7-220501096-AA5-EV02 API

Teniendo en cuenta lo realizado en la AA5-EV01 (evidencia anterior) realice el testing de la API construida usando la herramienta postman.

Elementos para tener en cuenta:

- Debe instalar la herramienta postman.
- Debe realizar un video mostrando el testing de la API con la herramienta Postman.
- Documentar con pantallazos el testing realizado.
- Entregar los ENDPOINT de las API´s

Lineamientos generales para la entrega de la evidencia:

- **Productos para entregar:** carpeta comprimida que debe tener los siguientes archivos: o Archivos del proyecto,

o Video
o Documento con pruebas.
o Archivo END POINTS.
o Archivo con enlace del repositorio, la carpeta comprimida debe tener el nombre del aprendiz y número de la evidencia así: NOMBRE_APELLIDO_AA5_EV02
o **Extensión:** ZIP, RAR.

- Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio: API GA7-220501096-AA5-EV02.

Evidencia de conocimiento: GA7-220501096-AA4-EV02 verificación de procedimientos para la definición de componentes frontend de la aplicación (listas de chequeo)

2. LISTA DE CHEQUEO

No.	VARIABLES/INDICADORES DE LOGRO	CUMPLE		Observaciones
		SI	NO	
1.	Realiza el test de la API utilizando POSTMAN			25%
2.	Realiza el video solicitado mostrando las pruebas del API			25%
3.	Realiza la documentación del API realizada			25%
4.	Entregar los ENDPOINT de las API´s desarrolladas			25%

2. Codificación de la API REST con Node.js

1 Buenas prácticas en la codificación de Node.js



2 Desarrollo de la API REST



Testing de una API REST con Postman

1. ¿Qué es Postman?

- Es una herramienta utilizada para **probar APIs REST**.
- Permite enviar solicitudes HTTP como: **GET, POST, PUT, DELETE, PATCH**, etc.
- Facilita la **automatización de pruebas**, la **documentación** y el **manejo de colecciones**.
- Muy usado por desarrolladores, QA testers y equipos DevOps.

2. ¿Qué es una API REST?

Una **API REST** es un servicio que permite la comunicación entre aplicaciones usando el protocolo HTTP.

3. Beneficios de usar Postman para pruebas de API

4. Tipos de pruebas que puedes hacer en Postman

✓ Pruebas manuales

✓ Pruebas automatizadas

✓ Pruebas de integración

✓ Pruebas de regresión

Testing de una API REST con Postman

5. Elementos clave en Postman

1. **Colecciones**
2. **Environments (Entornos)**
3. **Request (Solicitud)**
4. **Test Scripts**

6. Pruebas básicas que debes realizar a una API en Postman

Tipo de prueba	¿Qué se valida?
Status Code	200, 201, 400, 404, 500…
Estructura JSON	Campos esperados
Headers	Content-Type, Authorization
Tiempo de respuesta	< 1s o según SLA
Autenticación	Tokens, API Keys, JWT
Errores	Respuestas de error controladas

7. Ejemplo de Prueba en Postman — GET

Solicitud GET https://api.ejemplo.com/usuarios

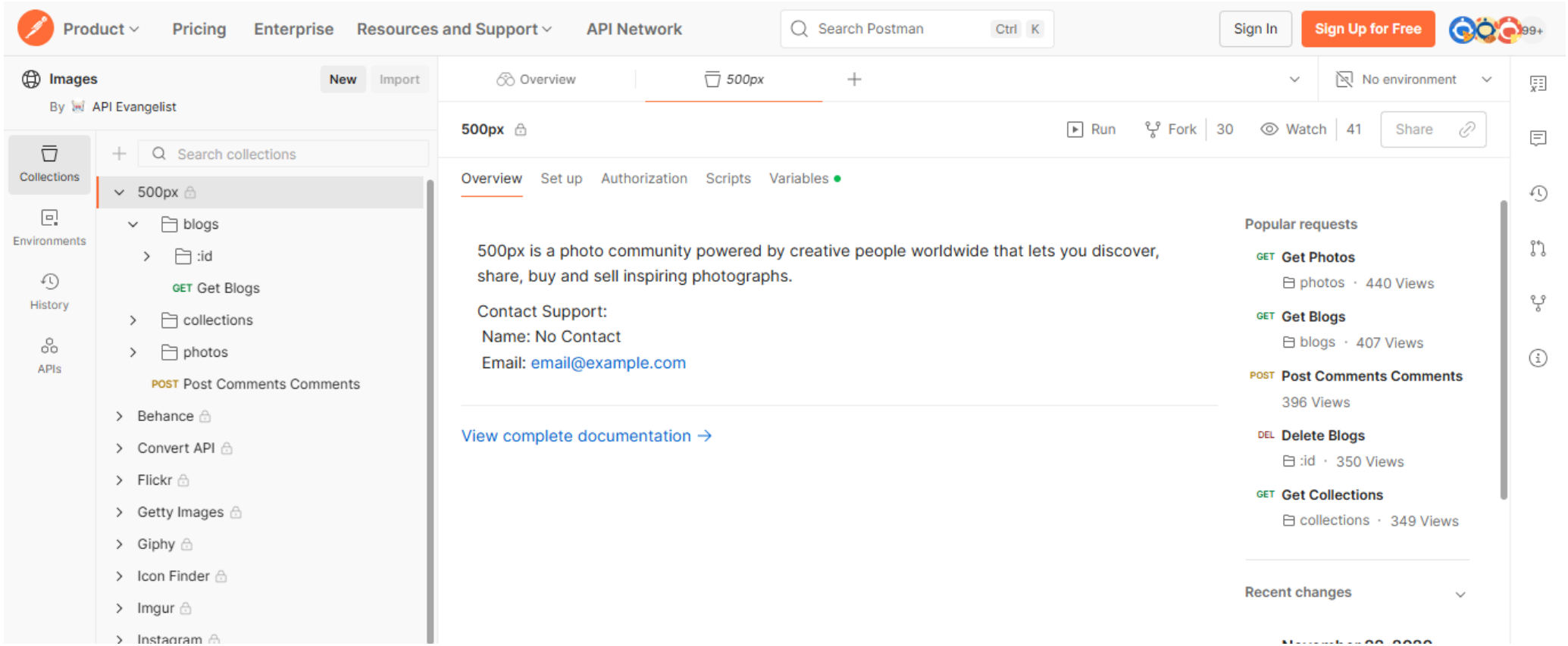
Script de prueba

```
pm.test("Status es 200", () => pm.response.to.have.status(200));
pm.test("El cuerpo tiene usuarios", () => {
    const body = pm.response.json();
    pm.expect(body).to.be.an("array");
});
```

Testing de una API REST con Postman

8. Testing Automatizado con Collection Runner

9. Buenas prácticas para probar APIs



10 Primeros Pasos en Pruebas de API desde Cero con Postman https://www.youtube.com/watch?v=wu5XOzAXNFE

3. Testing de la API REST con Postman

es una herramienta que se utiliza especialmente para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de tecnologías.



Para finalizar, conozca la instalación de la herramienta Postman.

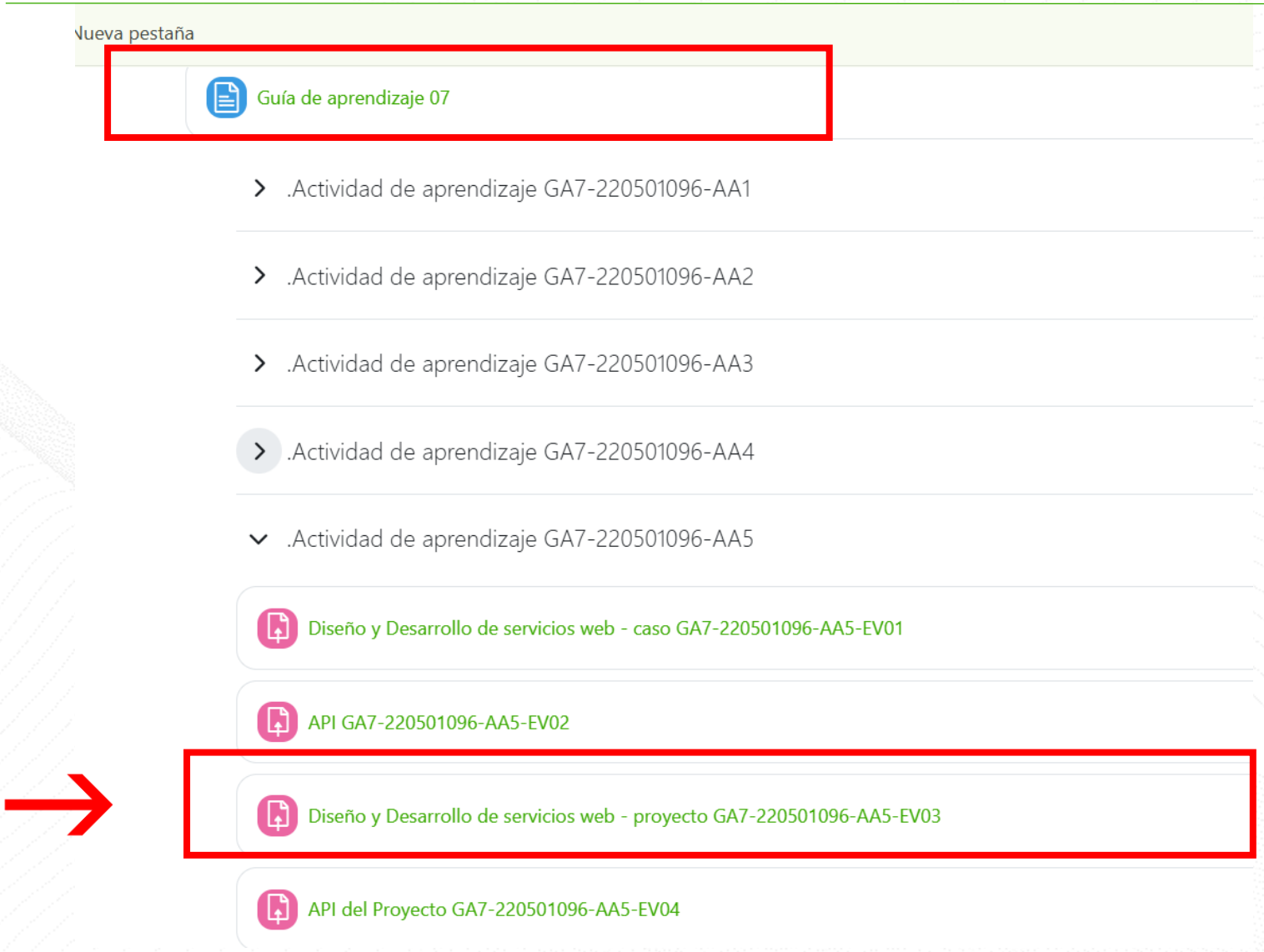


Video: Cómo Automatizar Pruebas a APIs desde Cero | Postman

https://www.youtube.com/watch?v=PvAsMgbRs5w

GA7-220501096-AA5-EV03: Diseño y desarrollo de servicios web - proyecto

Evidencia a desarrollar



Actividad de Proyecto 7

Material de formación

- Aplicación del paradigma orientado a objetos
- Integración continua
- Construcción de aplicaciones con JAVA
- Frameworks para construcción de aplicaciones con JAVA
- Desarrollo de frontend con React JS
- Construcción de API
- Servicios web con PHP
- Construcción aplicación web

Evidencia de desempeño: GA7-220501096-AA5-EV03: Diseño y desarrollo de servicios web - proyecto

Tomando como referencia lo visto en el componente formativo “Construcción de API” realizar el diseño y la codificación de las API’s teniendo en cuenta las características del software a desarrollar.

Elementos para tener en cuenta:

- Se requiere realizar los servicios necesarios para cumplir con las características del software a realizar para el proyecto formativo.
- Se debe crear la documentación de cada uno de los servicios.
- Se debe crear el proyecto utilizando herramientas de versionamiento.

Lineamientos generales para la entrega de la evidencia:

- **Productos para entregar:** carpeta comprimida que debe tener los siguientes archivos: ○ archivos del proyecto, ○ archivo con enlace del repositorio, la carpeta comprimida debe tener el nombre del aprendiz y número de la evidencia así: NOMBRE_APELLIDO_AA5_EV03
  - ○ **Extensión:** ZIP, RAR.
- Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio: diseño y desarrollo de servicios web – proyecto GA7-220501096-AA5-EV03.

Evidencia de desempeño: GA7-220501096-AA5-EV03: Diseño y desarrollo de servicios web - proyecto

2. LISTA DE CHEQUEO

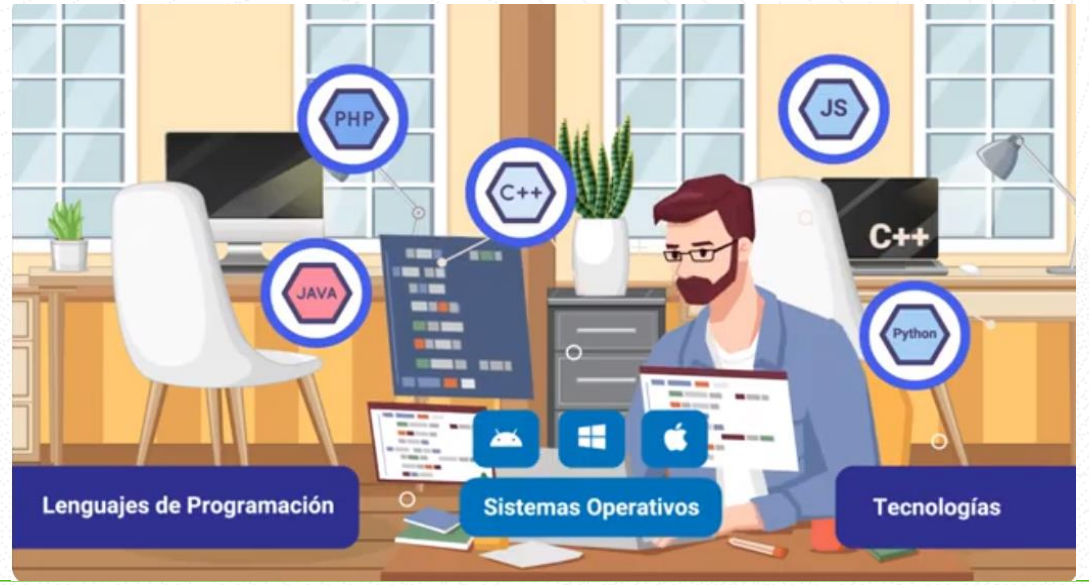
No.	VARIABLES/INDICADORES DE LOGRO	CUMPLE		Observaciones
		SI	NO	
1.	Realiza los servicios según requerimientos del proyecto			25%
2.	Realiza API´s Rest según necesidades del proyecto			25%
3.	Realiza las validaciones de verificación correctamente			25%
4.	Utiliza herramientas de versionamiento para la creación de proyecto			25%

1. Servicios web con PHP









1. Servicios web con PHP



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.



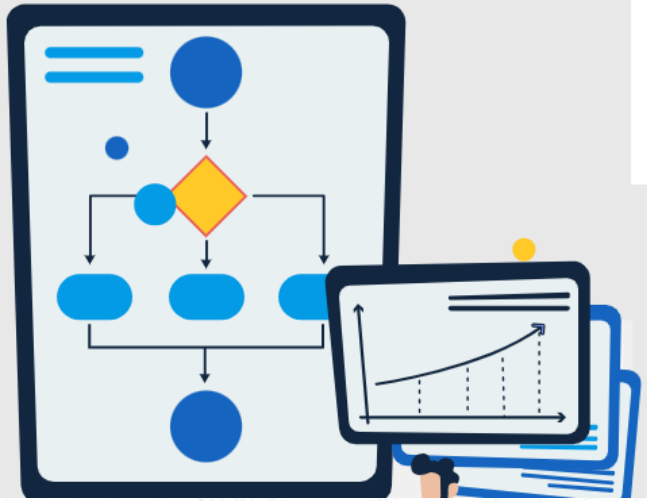
Ejemplo 1:

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  <?php
2  function media_aritmetica($a, $b){ //Aquí definimos la función
   y la llamamos media_arimetica y las variables a y b
3  $media=($a+$b)/2; //escribimos la fórmula de la media, en
   este caso de 2 números divididos entre 2
4  return $media; // aquí la función retorna el valor de la media
5  }
6  echo media_aritmetica(7,9),"<br>"; // aquí ingresamos los
   números, en este caso 7 y 9
7  echo media_aritmetica(300,500),"<br>"; ?> //aquí ingresamos
```

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 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.

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:



Una clase

Es como el plano de un apartamento, este 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

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

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

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.

Ejemplo

```
1  <?php
2          class Persona // Se define la clase Persona
3          {
4
5              private $nombre; // Declaración del campo. Este
debe ser de carácter privado.
6
7    //Aquí adelante se utiliza un constructor, que es un método
que lleva el nombre de la clase y que será el responsable de
construir el objeto.
8              function Persona($nom) {
9                  $this->nombre = $nom;// this hace referencia a
la instancia actual, y se le asigna el valor al constructor->
10
```



http://localhost/Php1/index.php

Objetos en PHP | Objetos en PHP

Objetos en PHP

El objeto objPersona se llama Mario Meneses

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

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 lenguaje 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.

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:

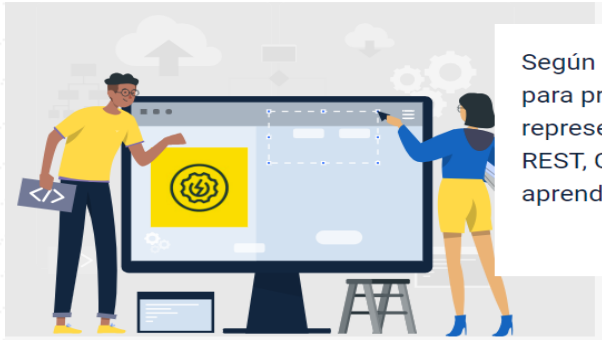
1. (DDL) Lenguaje de Definición de Datos
2. (DML) Lenguaje de Manipulación de Datos
3. (DCL) Lenguaje de Control de Datos

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".

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.

SOAP UI

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

Descargar

Plazo Actividades:

19 al 30 de Noviembre



G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682

@SENAComunica

www.sena.edu.co