PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

Guía Pruebas I.

- **Denominación del Programa de Formación**: TECNICO EN PROGRAMACION DE SOFTWARE
- **Código del Programa de Formación**: 233104 v1
- **Nombre del Proyecto**: CODIFICACION DE APLICACIONES DE SOFTWARE PARA EL SECTOR EDUCATIVO Y EMPRESARIAL
- **Fase del Proyecto**: PLANEACION
- **Actividad de Proyecto**: REALIZACION DE PRUEBAS UNITARIOAS PARA LA VERIFICACION DEL SOFTWARE
- **Competencia**: DESARROLLAR LA SOLUCION DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGIA DE DESARROLLO
- **Resultados de Aprendizaje Alcanzar**: VERIFICAR LA FUNCIONALIDAD DEL SOFTWARE APLICANDO PRUEBAS UNITARIAS
- **Duración de la Guía**

80 horas presenciales

2. PRESENTACION

La presente guía de aprendizaje orienta el aprendiz en temáticas que le permiten de acuerdo con su proyecto formativo, poder elaborar las pruebas del sistema de información, y documentarlas, detectando los errores en el sistema de información y teniendo en cuenta los requisitos documentados del cliente.

Para esta etapa en particular la actividad está relacionada con las técnicas de pruebas sobre el software que ayudan a elevar la productividad tanto en labores de desarrollo como en la misma calidad del producto software.

Antes de entregar un producto software al usuario final o cliente es necesario hacer pruebas con la intención de detectar errores y fallos de la aplicación y levantar la documentación necesaria, ya que de las experiencias aprendidas se pueden aplicar a otros trabajos mejorando así la productividad en la empresa y muestra madurez al interior de la misma.

Es importante comentar que este proceso no se hace en una fase determinada de la producción, al contrario, por ser una parte de la calidad se debe hacer durante todo el proceso. Las pruebas no aseguran la calidad, pero si permiten evaluarla y proporcionan retroalimentación a todos los integrantes del equipo; con esta documentación y experiencia con el tiempo se detectarán los errores de manera fácil y de manera efectiva en cuanto a tiempo y costo.

Se sugiere que primero lea la guía de aprendizaje en su totalidad, para que de manera ordenada y manejando los tiempos sugeridos, cumpla con las evidencias se inicie el desarrollo de cada una de las acciones propuestas. Le invitamos a participar activamente de las actividades y abrir su mente al conocimiento de nuevas tecnologías. Adelante y muchos éxitos en este propósito.

3. FORMULACION DE LAS ACTIVIDADES DE APRENDIZAJE

- Descripción de la(s) Actividad(es)

a. Pregunta Problematizadora: (individual)
Socializar en un debate la siguiente pregunta

Pregunta Problemica:

A partir del video de reflexión "1. Video_pruebas_de_software.mp4" conteste en el foro dispuesto en la plataforma los siguientes interrogantes:



.

- ¿Conoce al menos dos casos donde el software haya fallado y esta falla haya cobrado vidas o haya ocurrido un desastre informático?
- ¿Porque cree usted que son importantes las pruebas de software del sistema de información y el aseguramiento de la calidad?
- ¿Cómo elaborar unas buenas pruebas de software y asegurar la calidad del mismo?
- ¿Que debe tener en cuenta para elaborar las pruebas del software que hace parte de un sistema de información?

Es importante que el aprendiz realice un ejercicio exploratorio de evaluación, respondiendo a sí mismo las preguntas planteadas, y determinando sus fortalezas y aptitudes frente al tema, así como, la importancia de avanzar en la adquisición de competencias requeridas durante su proceso de aprendizaje para un necesario dominio del tema.

Actividad I:

La actividad de reflexión inicial ha generado inquietudes, y muy seguramente, suscitado alternativas para **Desarrollar las tareas de configuración y puesta en marcha del sistema de información** que en este momento el aprendiz proyecta realizar, con el ánimo de brindar una solución específica a una necesidad o problema en particular; esta posibilidad de introspección, abre el camino hacia un espacio de contextualización en el que se acomete la identificación y rescate de los saberes que el aprendiz ha construido a lo largo de sus particulares vivencias y desempeño laboral.



- Teniendo en cuanta el material de apoyo "**2. Documento pruebas software**" de acuerdo a los grupos de trabajo elabore utilizando la aplicación **Mindomo (**https://www.mindomo.com/es/) un mapa mental con los conceptos, tipos y procedimientos de pruebas de software,
- Realice el "Test – pruebas – software" para fortalecer los conocimientos de pruebas de software igualmente como herramienta de apoyo.

- Publicar las evidencias en la plataforma virtual de aprendizaje con el fin de que sean evaluadas por el instructor que apoya la ejecución del proceso;

Actividad II

Con el fin de obtener el aseguramiento de la calidad del sistema de información desarrollado y con base en el diseño de las pruebas de software basadas en los casos de prueba, se debe elaborar un documento donde se especifique el plan de pruebas y el registro del resultado de las mismas para ello el aprendiz deberá:

1. Elaborar laboratorio de pruebas de software: ***"4. Laboratorio de pruebas 1"*** en este paso se debe realizar los ejercicios propuestos en el laboratorio para apropiar los conceptos antes de elaborar el informe entregable.

Actividad Semana II:

Elaborar el plan de pruebas: en este paso se debe realizar una lectura y análisis del documento de requerimientos del proyecto, se debe tener construido el sistema de información, se debe elaborar el plan de pruebas con base a el documento "***5. Pruebas de software 2"*** para realizar la siguiente actividad:

a. Con base en los grupos conformados del proyecto realice el plan de pruebas para el proyecto formativo que se esta trabajando utilizando la plantilla ***"plantilla plan de pruebas".***
b. Diligencie y suba en la plataforma en el espacio habilitado por el instructor.

Actividad Semana III:

a. Realice mínimo 6 casos de prueba al proyecto formativo con base en la plantilla "plantilla caso de prueba" de acuerdo a los grupos conformados por el Instructor.
b. Una vez diligenciada las plantillas comprima en un archivo .rar ó .zip y suba en el espacio habilitado por el instructor en la plataforma



4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias De Conocimiento: a. Mapa mental Documento Pruebas de Software b. Test Pruebas de Software **Evidencias De Producto:** c. Informe Laboratorio Pruebas de Software1 d. Plantilla Plan de Pruebas e. Plantilla Casos de pruebas . Subir las anteriores evidencias a la plataforma	Realiza y documenta las pruebas del software según el plan establecido, para Definir acciones correctivas a seguir y asegurar el buen funcionamiento del Sistema de información.	● Búsqueda y análisis de información. ● Simulación de caso ● Aprendizaje basado en problemas ● Análisis discusión de grupos .

5. GLOSARIO DE TERMINOS

Prueba: Proceso mediante el cual se ejecuta de manera sistemática un conjunto de actividades (métodos y técnicas) para encontrar errores.
Caso de prueba: Conjunto de condiciones, datos o variables que servirán para determinar si los requisitos del sistema se cumplen de manera parcial, completa, o no se cumplen.
Error: Discrepancia entre el valor calculado y el valor teórico o esperado, con responsabilidad del desarrollador.
Defecto software: Desviación en el valor esperado por una cierta característica. Defecto de calidad.
Fallo: Consecuencia de un error o un defecto software.
Overflow: Significa desbordamiento en el buffer, cuando la cantidad de datos supera la capacidad preasignada. Es un fallo de programación.
Underflow: Significa subdesbordamiento del buffer, cuando se carga datos a una velocidad inferior a la de procesamiento, provocando bloqueos.

6. REFERENTES BIBLIOGRAFICOS

BOLAÑOS, D., SIERRA, A., & ALARCÓN, M. (2008). Pruebas de Software y JUnit. Madrid: Pearson Prentice Hall.

CATALDI, Z. (2000). Metodología de diseño, desarrollo y evaluación de software educativo. Tesis de Magíster en Informática. Argentina: Facultad de Informática. Universidad Nacional de la Plata (UNLP).

PRESSMAN, R. (2006). Ingeniería del Software: Un enfoque práctico. Sexta edición. McGrawHill.

IEEE Standard Glossary of Software Engineering Terminolgy [IEEE, 1990].

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Jhon Edward Vaquiro Leal	Instructor	Centro de industria y Construcción	10-2-2020

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	**Cargo**	**Dependencia**	**Fecha**	**Razón del Cambio**
Autor (es)	Equipo ingenieros de sistemas - Articulación	Instructores	Centro de Comercio y Turismo	Octubre de 2022	Actualización de código del programa

	SERVICIO NACIONAL DE APRENDIZAJE SENA **INSTRUMENTO DE EVALUACIÓN**

Regional: Quindío
Centro de formación : Comercio y Turismo

Programa de Formación: Técnico en Programación de software
Código del programa: 233104
Versión: **1**

Competencia a Desarrollar: Desplegar la solución de software

Resultado de Aprendizaje: Interpretar el diagrama para identificar el modelo de datos

Actividad de aprendizaje: Elaborar el manual de usuario, según normas y protocolos establecidos.

Evidencia:
Desempeño, juego de roles

Nombre del Aprendiz: _____ No. documento _____
Nombre del Instructor: _____ No. documento _____
Duración de la evaluación:

INSTRUCCIONES PARA EL DILIGENCIAMIENTO:

En el desarrollo de este instrumento se valorará las evidencias de conocimiento del proceso con base en las tecinas y habilidades desarrollados durante la formación, estas evidencias se valoran con base en las actividades que se describen en la guía de aprendizaje que fueron socializadas y concertadas con el instructor, con base en sus evidencias el instructor valorará cada uno de los factores que contiene las siguientes listas de chequeo para alcanzar el resultado de aprendizaje.

Crear un manual de usuario mediante un juego de roles

Actividad metodología scrum:

1. Formar grupos de máximo 5 personas.
2. Definir los roles (scrum master, product owner, team scrum).
3. Comenzar daily no mayor a 15 minutos.

Un CompromISO de Todos	**SERVICIO NACIONAL DE APRENDIZAJE SENA** **INSTRUMENTO DE EVALUACIÓN**

LISTA DE CHEQUEO

No.	Aspecto	INDICADORES/FACTORES A EVALUAR	CUMPLE		Observaciones
			SI	NO	
1	Actitud Reflexión	El aprendiz participó de manera activa y colaborativa en la actividad de reflexión			
2	Actitud Contextualización	El aprendiz participó de manera activa y colaborativa en la actividad de contextualización			
3	Actitud Apropiación	El aprendiz participó en las actividades didácticas para la apropiación del conocimiento			
4	Actitud Apropiación	El aprendiz se presenta puntual a las actividades de aprendizaje			
5	Actitud Apropiación	El aprendiz evidencia un buen porte del uniforme durante el proceso de aprendizaje			
6	Actitud Apropiación Técnico	El aprendiz realiza la elaboración del manual de usuario describiendo los conceptos básicos.			
8	Actitud Proyecto-transferencia	El aprendiz elabora el manual de usuario mediante la metodología scrum			
9	Actitud Portafolio de evidencias	El aprendiz registra las evidencias en su portafolio con base en las fechas e instrucciones dadas por el instructor			

RESULTADOS: Aprobado ☐ Deficiente ☐

OBSERVACIONES-RECOMENDACIONES

Ciudad y fecha de aplicación del instrumento:

_____ _____

Firma del Aprendiz Firma Instructor

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- ✓ Denominación del Programa de Formación: Tecnico en Programación de Software
- ✓ Código del Programa de Formación: 233104 Version 1
- ✓ Nombre del Proyecto: Codificación de aplicaciones de software para el sector educativo y empresarial del departamento del Quindío.
- ✓ Fase del Proyecto: Ejecución.
- ✓ Actividad de Proyecto: Realizar actividades de despliegue, elaboración de un plan de trabajo para realizar la verificación de la implantación del software de acuerdo a standares de la industria del desarrollo.
- ✓ Competencia: Planear las actividades de implantación del software de acuerdo con plataforma
- ✓ Resultados de Aprendizaje: 584230 - PLANEAR LAS ACTIVIDADES DE IMPLANTACIÓN DEL SOFTWARE DE ACUERDO CON PLATAFORMA TECNOLÓGICA.
- ✓ Duración de la Guía: 120 Horas

2. PRESENTACIÓN

Con esta Guía, se busca que los aprendices realicen las respectivas verificaciones de la implementación de un pag Web. Para realizar diferentes pruebas de rendimiento para comprobar el rendimiento del sitio Web se debe tener unos conocimientos teóricos sobre los tipos de pruebas que existen y cuando se debe implantar cada una de estas es necesario entender los tipos de pruebas funcionales y no funcionales.



Tipos funcionales: Las pruebas funcionales se llevan a cabo para comprobar las características críticas para el negocio, la funcionalidad y la usabilidad. Las pruebas funcionales garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema. Valida principalmente toda la aplicación con respecto a las especificaciones mencionadas en el documento Software Requirement Specification (SRS). Los tipos de pruebas funcionales incluyen pruebas unitarias, pruebas de interfaz, pruebas de regresión, además de muchas.

Pruebas unitarias: Las pruebas unitarias se centran en probar piezas/unidades individuales de una aplicación de software al principio del SDLC. Cualquier función, procedimiento, método o módulo puede ser una unidad que se someta a pruebas unitarias para determinar su corrección y comportamiento esperado. Las pruebas unitarias son las primeras pruebas que los desarrolladores realizan durante la fase de desarrollo.

Pruebas de integración**:** Las pruebas de integración implican probar diferentes módulos de una aplicación de software como grupo. Una aplicación de software se compone de diferentes submódulos que trabajan juntos para diferentes funcionalidades. El propósito de las pruebas de integración es validar la integración de diferentes módulos juntos e identificar los errores y problemas relacionados con ellos.

Pruebas no funcionales: Las pruebas no funcionales son como pruebas funcionales; sin embargo, la principal diferencia es que esas funciones se prueban bajo carga para el rendimiento de los observadores, fiabilidad, usabilidad, escalabilidad, etc. Las pruebas no funcionales, como las pruebas de carga y esfuerzo, normalmente se llevan a cabo mediante herramientas y soluciones de automatización, como LoadView. Además de las pruebas de rendimiento, los tipos de pruebas no funcionales incluyen pruebas de instalación, pruebas de confiabilidad y pruebas de seguridad.

Performance Testing**:** Las pruebas de rendimiento son un tipo de pruebas no funcionales, realizadas para determinar la velocidad, estabilidad y escalabilidad de una aplicación de software. Como su nombre indica, el objetivo general de esta prueba es comprobar el rendimiento de una aplicación con respecto a los diferentes puntos de referencia del sistema y de la red, como la utilización de la CPU, la velocidad de carga de la página, el control de tráfico máximo, la utilización

de recursos del servidor, etc. Dentro de las pruebas de rendimiento, hay varios otros tipos de pruebas, como pruebas de carga y pruebas de esfuerzo.

Cuál es la diferencia en que existe entre estas dos pruebas Supongamos que tiene un sitio web/aplicación de comercio electrónico con funcionalidades estándar. Estos son algunos ejemplos de pruebas de rendimiento, pruebas funcionales, pruebas de integración y pruebas unitarias:

Ejemplo relacionado aplicado en portales web Si desea comprobar cómo funcionará su sitio web cuando un alto número de usuarios acudan a su sitio web, por ejemplo, durante la temporada de ventas, debe realizar pruebas de carga, que entran dentro de la categoría de pruebas de rendimiento. Le ayudará a detectar problemas de velocidad y estabilidad y eliminar posibles cuellos de botella de rendimiento. Para este tipo de validaciones se hace necesario generar una lista de chequeo esto con el fin de verificar todos los item que se requiera verificar.

¿Qué es un checklist y cómo se debe utilizar?



Los listados de control, listados de chequeo, checklist u hojas de verificación, siendo formatos generados para realizar actividades repetitivas, controlar el cumplimiento de un listado de requisitos o recolectar datos ordenadamente y de manera sistemática. Se utilizan para hacer comprobaciones sistemáticas de actividades o productos asegurándose de que el trabajador o inspector no se olvida de nada importante.

Para qué sirven las listas de chequeo
Los principales de los checklist son los siguientes:

- ✓ Durante la realización de actividades en las que es muy importante que no se olvide ningún paso y deben hacerse las tareas con un orden establecido.
- ✓ Realizar inspecciones donde se deja constancia de cuales han sido los puntos inspeccionados.
- ✓ Verificar o examinar artículos, productos. En este caso, el SOFTWARE.
- ✓ Examinar o analizar la localización de los defectos. Verificando las causas de los defectos.

- ✓ Verificar y analizar las operaciones.
- ✓ Recopilar datos para su futuro análisis.
- ✓ El listado suele ser utilizado para realizar las comprobaciones rutinarias y asegurar que al operario o el encargado de dichas comprobaciones no se le pasa nada por algo, además de que se realice la simple obtención de datos.

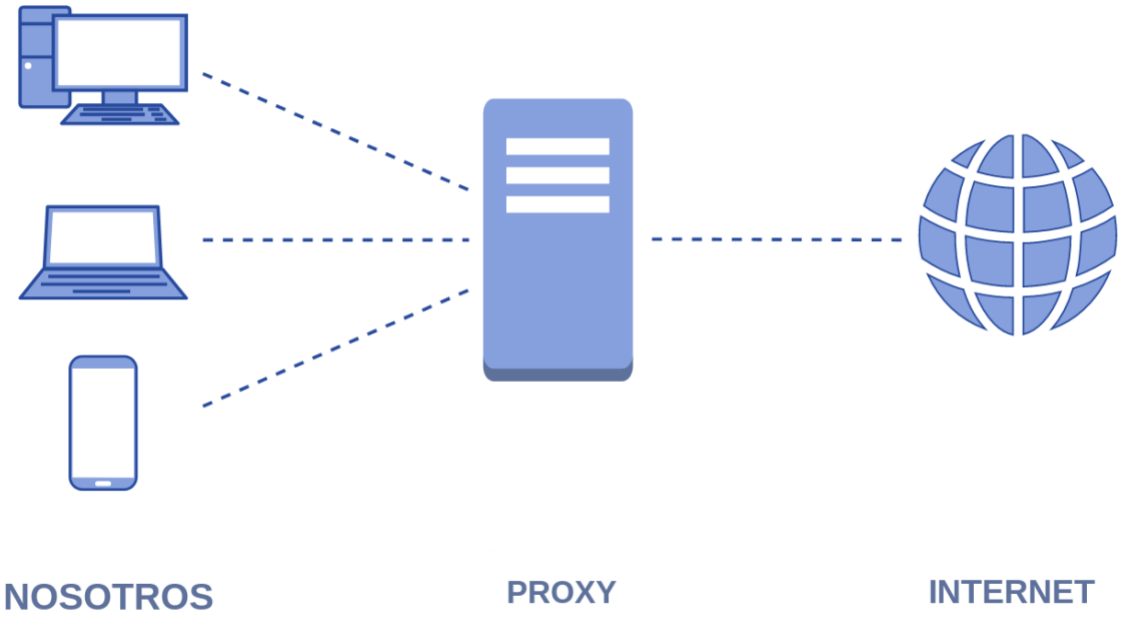
- ✓ La ventaja de los checklist es que, además de sistematizar todas las actividades que se deben realizar, una vez que se han rellenado sirven de registro, y puede ser revisado de manera posterior para tener constancia de lasdiferentes actividades que se realizan en un momento dado.

- ✓ Un checklist es una herramienta de ayuda en el trabajo que se diseña para reducir los errores provocados por los potenciales límites de la memoria y la atención en el ser humano. Ayuda a asegurar la consistencia.

- ✓ exhaustividad en la realización de una tarea. Un ejemplo sencillo de un listado de comprobación será un listado de tareas pendientes.

3. Servidor Web

Un servidor hace referencia a un proveedor de servicio, pueden ser uno o varios dispositivos como computadores generalmente de gran potencia, dedicadas a brindar un servicio. Sin embargo un servidor también puede tratarse de un software que nos brinde un servicio como por ejemplo un servidor Proxy el cual puede ser un servidor físico o un software que se ejecuta dentro un computador.



Ejemplo de un servidor web, Facebook tiene servidores donde se guardan los datos de los perfiles de los usuarios creados.

Las terminales como dispositivos finales se les llama Host Se traduce como anfitrión, son los elementos que componen una red, y hacen uso de está, tales como computadores, teléfonos, o consolas de video juegos. Cualquier componente que se conecte a una red es un host, incluso un servidor puede es un host.

¿Como funciona una pag. web?

El primer paso consiste en traducir la parte nombre del servidor del URL en una dirección IP usando la base de datos distribuida de Internet conocida como DNS. Esta dirección IP es necesaria para contactar con el servidor web y poder enviarle paquetes de datos.

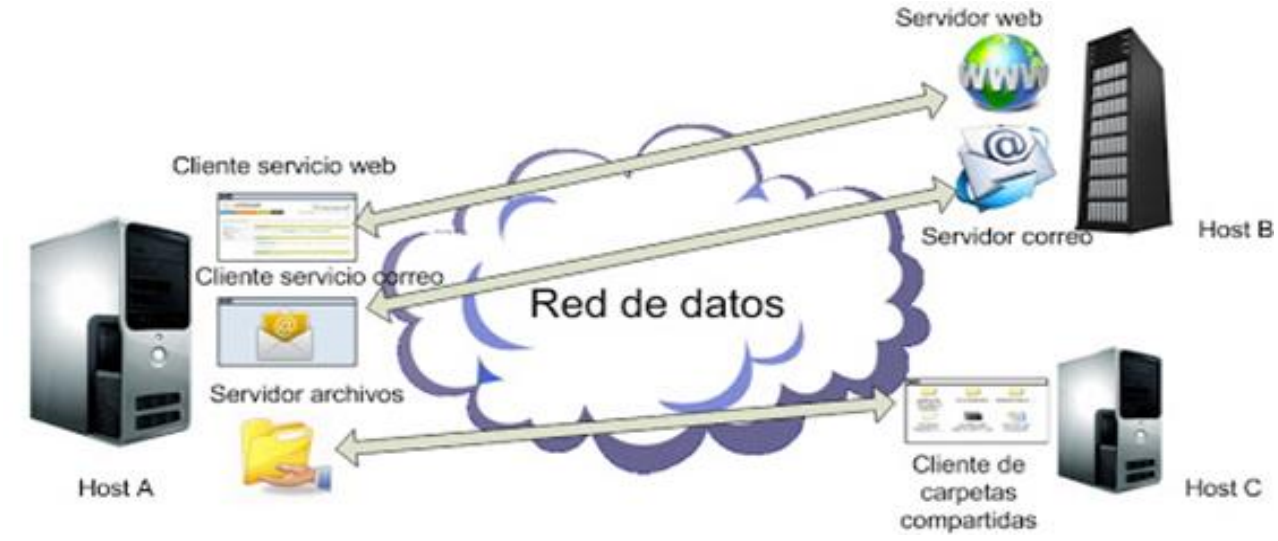
El siguiente paso es enviar una petición HTTP al servidor web solicitando el recurso. En el caso de una página web típica, primero se envía el texto HTML y luego el mismo es

inmediatamente analizado por el navegador, el cual, después, hace peticiones adicionales para los gráficos y otros ficheros que formen parte de la página. Las estadísticas de popularidad de un sitio web normalmente están basadas en el número de páginas vistas o peticiones de servidor asociadas, o peticiones de fichero, que tienen lugar.

Al recibir los ficheros solicitados desde el servidor web, el navegador representa (renderiza) la página tal y como se describe en el código HTML, el CSS y otros lenguajes web. Al final se incorporan las imágenes y otros recursos para producir la página que ve el usuario en su pantalla.

Modelo cliente – servidor

El modelo cliente servidor es un método de diseño de software en el cual se divide las tareas entre dos partes. Consiste en un cliente que no es que un software que se ejecuta en un host y este se conecta a través de una red a un servidor y realizar una petición, el servidor procesa esta petición y brinda una respuesta al cliente.



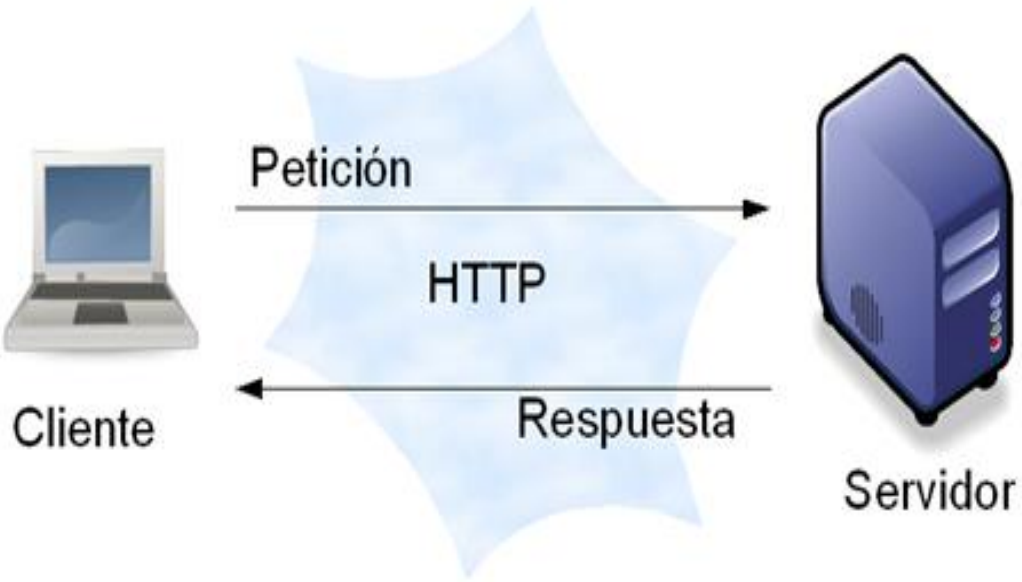
En este protocolo existen una serie de conceptos tales como:

- ✓ Conexión: es el circuito virtual establecido entre 2 programas en una red de comunicación

- ✓ Mensaje: es la unidad básica de un protocolo HTTP y consiste en una secuencia estructurada que se transmite entre los programas

- ✓ Cliente: es el programa que hace la llamada al servidor y es el que atiende en la transmisión la trama de los mensajes

- ✓ Servidor: es el programa que presta el servicio en la red.



Proxy: se trata de un programa intermedio cuya función puede ser variada, puede desde controlar el acceso desde ciertas direcciones, como filtrar contenido proviniendo de algunas direcciones.

Elementos que interviene en la comunicación:

- ✓ Conexión: establecimiento de una conexión del cliente con el servidor.
- ✓ Solicitud: envío por parte del cliente de un mensaje de solicitud al servidor.
- ✓ Respuesta: envío por parte del servidor de una respuesta al cliente.
- ✓ Cierre: fin de la conexión por parte del cliente y el servidor.

4. Servidores DNS

Es un software que se encarga de traducir la URL una dirección IP es decir cuando escribirnos https://www.youtube.com/ el servidor DNS lo traduce a una dirección como la siguiente https://208.65.153.238/.

Esto es lo que permite que los enrutadores de internet puedan encaminar el paquete por la internet y este encuentre el destinatario.

Se bebe tener presente que esto es posible mediante el protocolo TCP-IP

- ✓ Al colocar una URL en nuestro navegador (el cliente) realiza una petición a un servidor DNS para que le traduzca el nombre del dominio en una dirección IP
- ✓ Una vez se tenga la dirección IP del servidor donde se aloja la página que se desea visitar, el navegador envía un mensaje (una petición) mediante el protocolo HTTP. En este caso la petición es que le envié los datos de la página WEB que desea visitar



el servidor procesa esa petición y enviar los datos de la página web que el cliente solicitó, una vez más por mediante el protocolo HTTP al llegar la información del servidor nuestro navegador(el cliente) toma esa información y realiza la representación de la página web.

En otras palabras, el servidor lo que hace es enviar las instrucciones para que nuestro navegador cree una copia de cómo se debería ver el sitio web que estamos visitando.



La URL (Uniform Resource Locator) o Localizador Uniforme de Recursos, es la dirección específica que se le asigna a un servicio (servidor) disponible en la WEB para que este sea localizado, lo que quiere decir que hay una URL para cada uno de los recursos (páginas, sitios, documentos, archivos, carpetas) que hay en la World Wide Web.

Una URL como https://www.youtube.com/ está compuesta por varias partes:

- ✓ 'https://' que sería el protocolo de acceso para las páginas de internet. Otro ejemplo es el 'ftp://' que es el protocolo para descarga de ficheros.

- ✓ 'www.' que es la red donde se encuentra.

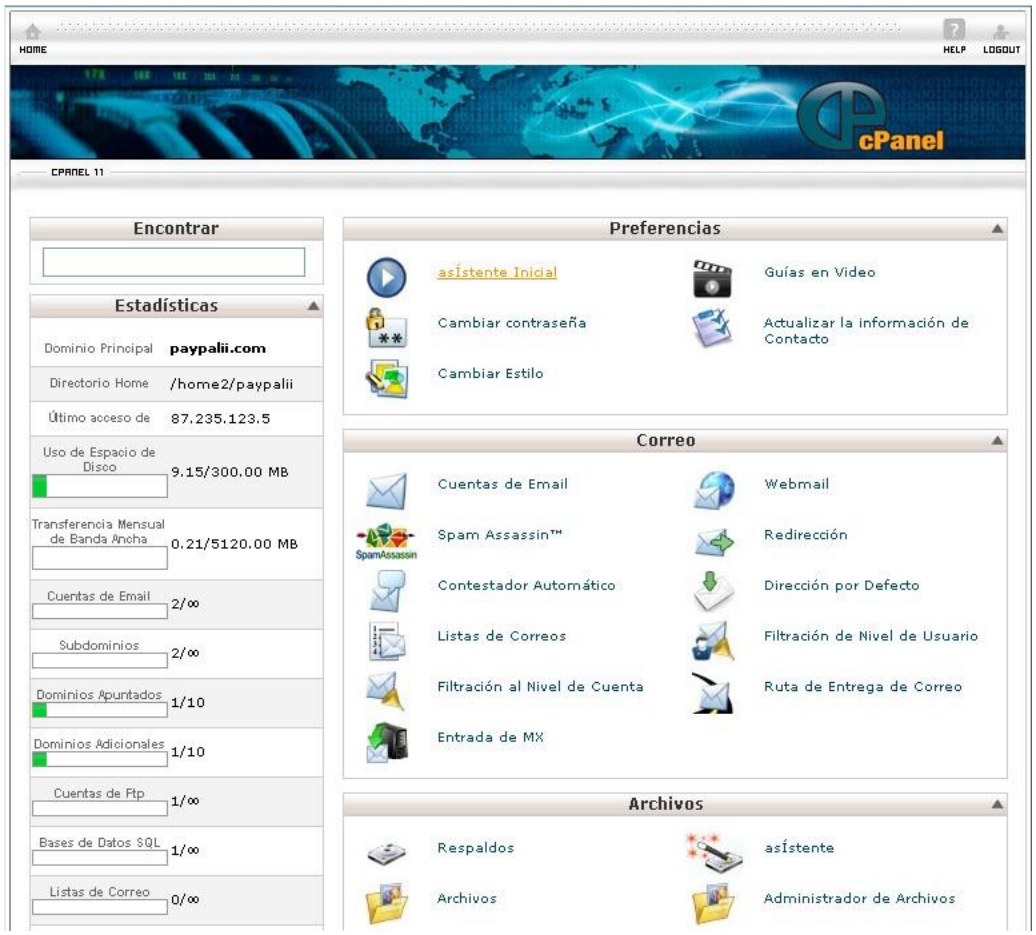
- ✓ ‘youtube’ sería el nombre del dominio

- ✓ '.com' es el tipo de dominio. Estos pueden ser genéricos como .net, .org, .mobi o territoriales como .mx, .ar, .cl. Hoy en día el tipo de dominio se está diversificando pudiente contener frases completas como .google o .maps.

http://www.eie

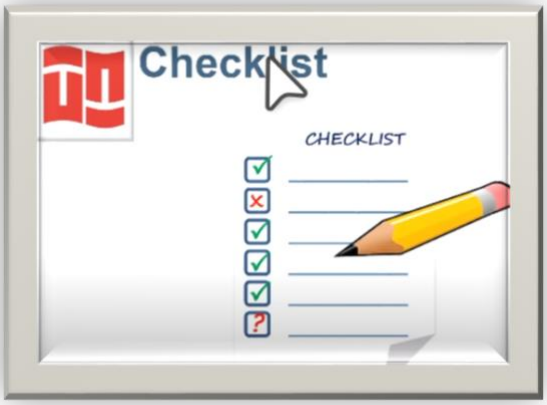
5. Cpanel

El cpanel el panel principal para administrar servicios de un servidor web entre ellos elementos como: Preferencias, Correo, Archivos, Registros, Seguridad, Dominios, Bases de Datos, Software/Servicios y Avanzado.



GUIA4 Proyecto Integrador: Actividad 1 PERFORMANCE TESTING

Objetivo: Cargar la pag. Web en un hosting gratuito con el fin validar la presentación en la misma y verificar el comportamiento de la pag. Web mediante la empleando para ello creamos una cuenta en www.byethost.com y utilizamos el cpapel para configurar nuestra pag. Web para esto debemos cargar las carpetas de hdocs en Online File Manager como se indica en las siguientes imágenes una cargada la pag debemos subir nuestra BD en nuestro servidor web de alojamiento gratuito esto



con el fin de verificar los siguientes item:

- **Evalúa la velocidad y escalabilidad del sitio web/aplicación.**
- **Detecte si hay errores que se pasan por alto en las pruebas funcionales.**
- **Verifique la fiabilidad del sitio web bajo una gran carga.**

Imagen 1: cargue los archivos de su pag. Web como se describe en la imagen.

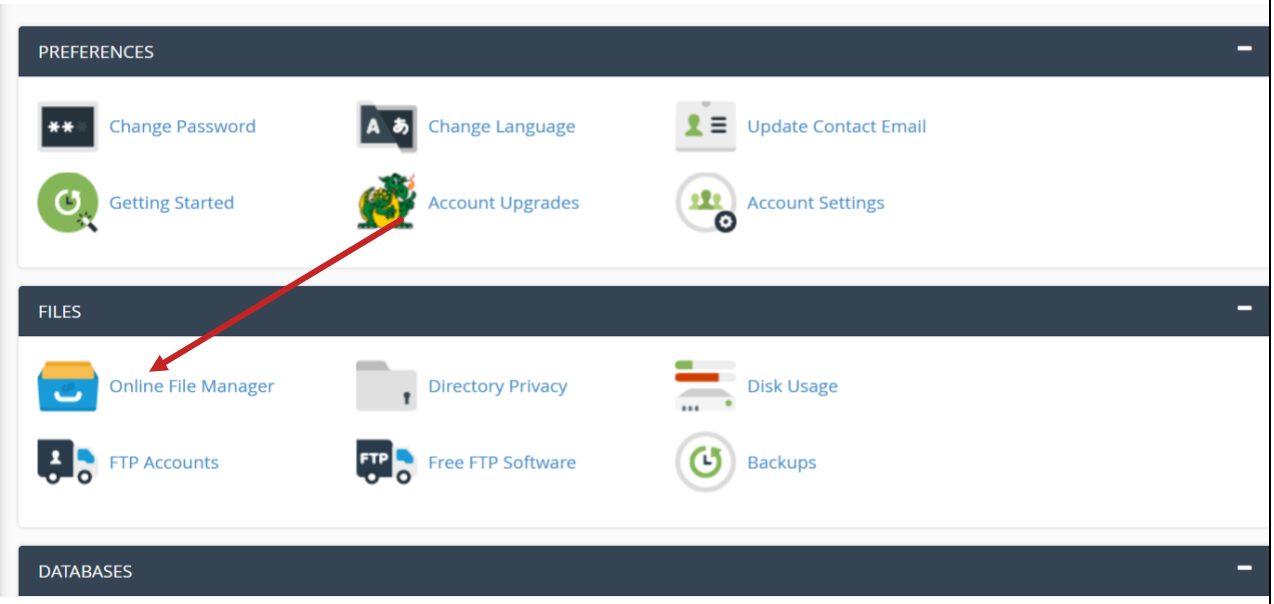


Imagen 2: Cargue los archivos dentro en la carpeta hdocs



Imagen 3: carge la base de datos de contiene la pag. Web

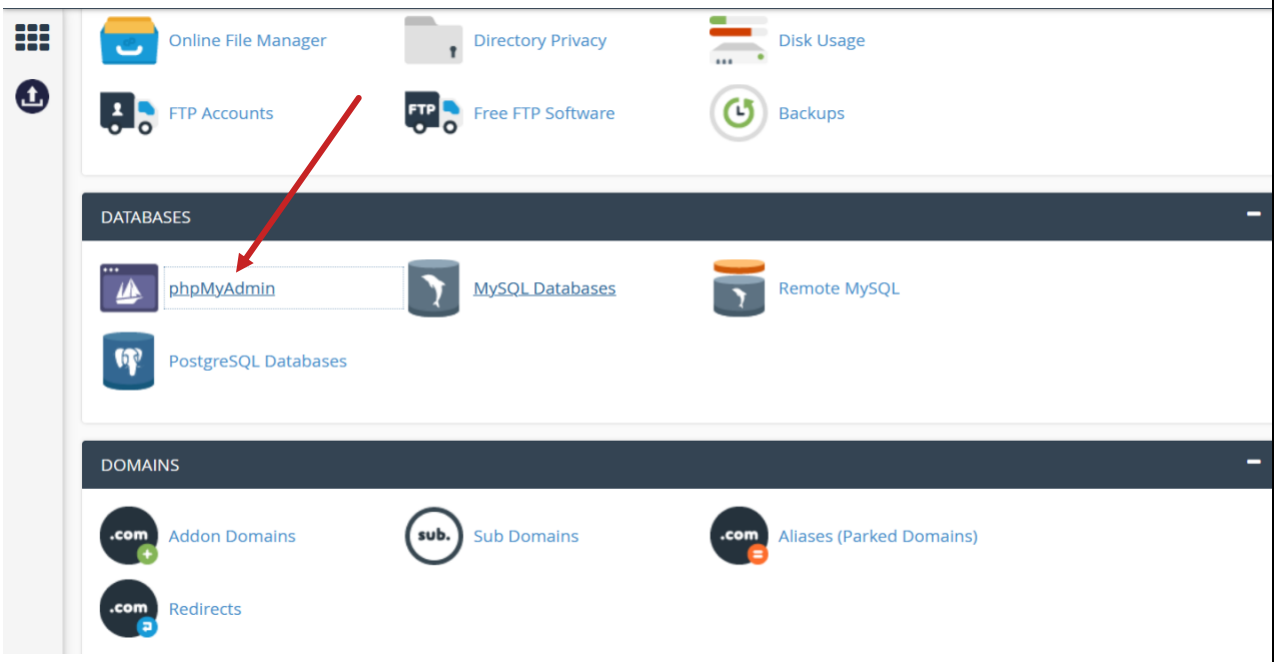
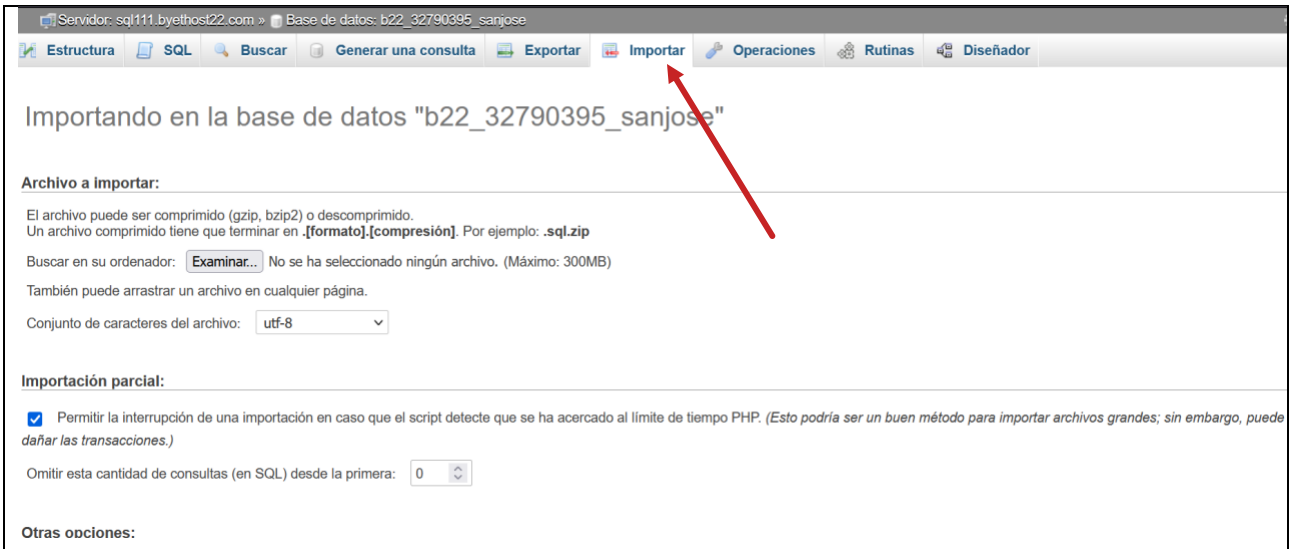
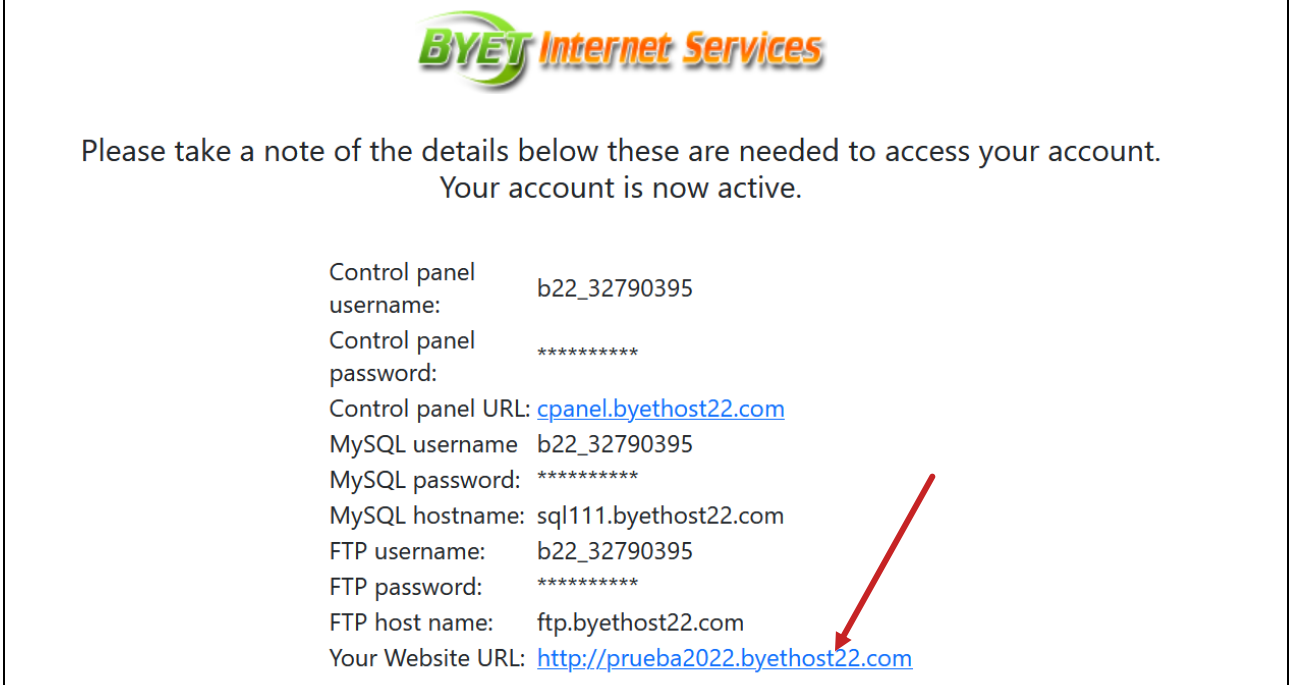


Imagen 4: Importar la BD que tiene relación con la pag. Web



Una ver cargada la BD verifique el carge y funcionalidad correcta de la pag. Web teniendo presente la URL que le idicaron al momento de crear su cuenta.

Imagen 5: Entra la URL de la pag. Web.



Evidencia: Etiquetar el archivo en PDF: "Actividad 1_Nombre del aprendiz"

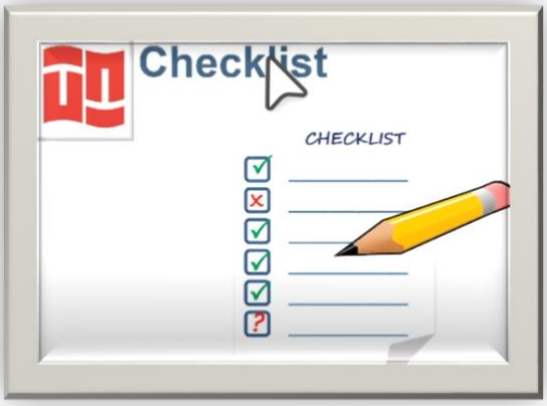
GUIA 4 PROYECTO INTEGRADOR: ACTIVIDAD 2 PRUEBAS FUNCIONALES

- **Se asegura de que el sitio web / aplicación está libre de defectos.**



Garantiza el comportamiento Verificar los siguientes Item:

- **Verifique se esperan los resultados esperados y todas las funcionalidades de la pág. Web**
- **Verifique que la arquitectura sea correcta con la seguridad necesaria si está manejando loguearse para accedes ha información confidencial.**



Objetivo: Hacer una liste de chequeo con el fin de verificar de generar pruebas funcionales en la pg. web

Evidencia: Etiquetar el archivo en PDF: "Actividad 2_Nombre del aprendiz"

GUIA2 PROYECTO INTEGRADOR: ACTIVIDAD 3_PRUEBAS DE INTEGRACIÓN

- **Verificar que los módulos de aplicación estén bien integrados y funcionen juntos según lo esperado.**
- **Identifique si problemas y conflictos interconectados para resolverlos antes de crear un gran problema.**
- **Valide si la funcionalidad, fiabilidad y estabilidad entre diferentes módulos.**
- **Detecte excepciones ignoradas para mejorar la calidad del código.**



Objetivo: Hacer una liste de chequeo con el fin de verificar de generar pruebas de integridad en la pag. web

Evidencia: Etiquetar el archivo en PDF: "Actividad 3_Nombre del aprendiz"

4. ACTIVIDADES DE EVALUACIÓN

Tome como referencia la técnica e instrumentos de evaluación citados en la guía de Desarrollo Curricular

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Producto: Actividad 1_Instalaccion crear una cuenta gratis en servidor web que Cpanel	Planear las actividades de implantación del software de acuerdo con plataforma	**Elaborar una página web basica en PHP, página web que contenga elementos de Html, Php, Mysql, Servidor web, Cpanel y hacer la verificacion de los siguientes Item:** **Evalúa la velocidad y escalabilidad del sitio web/aplicación.** **Detecte si hay errores que se pasan por alto en las pruebas funcionales.** **Verifique la fiabilidad del sitio web bajo una gran carga.**
Actividad 2_ pruebas funcionales		Verificar mediante una lista de chequeo la funcionalidad de la pág. Web teniendo en cuenta las consulta que se relaciones entre la BD y la pag. Web mediante la verificación de los siguientes Item: **Verifique se esperan los resultados esperados y todas las funcionalidades de la pág. Web.** **Verifique que la arquitectura sea correcta con la seguridad necesaria si está manejando loguearse para accedes ha información confidencial.**

Actividad 3_ *pruebas de integridad*		pruebas de integridad mediante una lista de cheque con el fin de verificar los siguientes Item: **Verificar que los módulos de aplicación estén bien integrados y funcionen juntos según lo esperado.** **Identifique si problemas y conflictos interconectados para resolverlos antes de crear un gran problema.** **Valide si la funcionalidad, fiabilidad y estabilidad entre diferentes módulos.** **Detecte excepciones ignoradas para mejorar la calidad del código.**

5. GLOSARIO DE TÉRMINOS

Gestor de bases de datos: Programa que permite crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

Lenguaje de definición de datos (DDL: Data (Definition Language): “Sencillo lenguaje artificial para definir y describir los objetos de la base de datos, su estructura, relaciones y restricciones.” (Hueso, I. L. 2014).

Lenguaje de control de datos (DCL: Data Control Language): “Encargado del control y seguridad de los datos (privilegios y modos de acceso, etc). Este lenguaje permite especificar la estructura y el tipo de los datos, asÌ como las restricciones sobre los datos. Todo esto se almacenar·en la base de datos.” (Hueso, I. L. 2014).

Lenguaje de manipulación de datos (DML: Data Manupulation Languaje): “Para la inserción, actualización, eliminación y consulta de datos. Para tal fin el lenguaje por excelencia es conocido SQL (Structured Query Language). Incluye instrucciones para DDL, DCL y DML y por su sencillez y potencia se ha convertido en el lenguaje est·ndar de los SGBD relacionales.” (Hueso, I. L. 2014).

Redundancia: Repetición innecesaria de datos.

Archivo: Es un elemento que contiene información.

Campo: Nombre que recibe cada columna de una tabla de una base de datos. Todos los datos que aparecen en un campo se refieren a un mismo tipo de información, sobre un objeto o individuo.

Claves o llaves: Conjunto de atributos que permiten identificar unívocamente a una entidad.

Entidad: Objetos principales en donde se almacena información.

Grado: Número de atributos de la tabla.

Integridad referencial: Propiedad deseable en las bases de datos relacionales que asegura que no haya datos incoherentes. Gracias a ella, se garantiza que una entidad, siempre se relacione con otras entidades válidas, es decir, las que existen en la base de datos.

Redundancia de datos: Presencia de datos repetidos en varios archivos.
Registro: Nombre que recibe cada fila de una tabla en una base de datos. Todos los datos que aparecen en un mismo registro se refieren a un mismo objeto o individuo. También se le conoce con el nombre de tupla o fila.

Tabla: Estructura básica sobre la que se construye una base de datos, ya que con ella se define qué información se va a almacenar y cómo se va a almacenar.

Vista: Es una relación virtual que no tiene por qué existir necesariamente en la base de datos, sino que puede producirse cuando se solicite por parte de un usuario concreto. de una tabla.

6. REFERENTES BILBIOGRÁFICOS

- ✓ ¿Qué es un checklist y cómo se debe utilizar https://www.isotools.org/2018/03/08/que-es-un-checklist-ycomo-se-debe-utilizar/

- ✓ Date, C. J. Ruiz FaudÛn, S. L. M. (2001). Introducción a los sistemas de bases de datos (7™ Ed.).
- ✓ MÈxico: Pearson Prentice Hall. F. Korth, H. Silberschatz, A. Sudarshan, S.(2006). Fundamentos de bases de datos (Vol. 5). Madrid: M cGraw -Hill.

- ✓ Cabello, G. J. M. (2014). Almacenamiento de la información e introducción a SGBD: administración de bases de datos (UF1468). Madrid, ESPA—A: IC Editorial. Recuperado de http://www.ebrary.com.bdigital.sena.edu.co, Hueso, I. L. (2014).

- ✓ Administración de sistemas gestores de bases de datos. Madrid, ES: RA-MA Editorial. Recuperado de http://www.ebrary.com.bdigital.sena.edu.co

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha

Autor (es)	Equipo ingenieros de sistemas - Articulación	Instructores	Comercio y Turismo	01-06-2022

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	**Cargo**	**Dependencia**	**Fecha**	**Razón del Cambio**
Autor (es)					

	SERVICIO NACIONAL DE APRENDIZAJE SENA **INSTRUMENTO DE EVALUACIÓN**

Regional: Quindío
Centro de formación : Comercio y Turismo

Programa de Formación: Técnico en Programación de software
Código del programa: 233104
Versión: **1**

Competencia a Desarrollar: Desplegar la solución de software

Resultado de Aprendizaje: Interpretar el diagrama para identificar el modelo de datos

Actividad de aprendizaje: Elaborar el manual de usuario, según normas y protocolos establecidos.

Evidencia:
Desempeño, juego de roles

Nombre del Aprendiz: _____ No. documento _____
Nombre del Instructor: _____ No. documento _____
Duración de la evaluación:

INSTRUCCIONES PARA EL DILIGENCIAMIENTO:

En el desarrollo de este instrumento se valorará las evidencias de conocimiento del proceso con base en las tecinas y habilidades desarrollados durante la formación, estas evidencias se valoran con base en las actividades que se describen en la guía de aprendizaje que fueron socializadas y concertadas con el instructor, con base en sus evidencias el instructor valorará cada uno de los factores que contiene las siguientes listas de chequeo para alcanzar el resultado de aprendizaje.

Crear un manual de usuario mediante un juego de roles

Actividad metodología scrum:

1. Formar grupos de máximo 5 personas.
2. Definir los roles (scrum master, product owner, team scrum).
3. Comenzar daily no mayor a 15 minutos.

Un CompromISO de Todos	**SERVICIO NACIONAL DE APRENDIZAJE SENA** **INSTRUMENTO DE EVALUACIÓN**

LISTA DE CHEQUEO

No.	Aspecto	INDICADORES/FACTORES A EVALUAR	CUMPLE		Observaciones
			SI	NO	
1	Actitud Reflexión	El aprendiz participó de manera activa y colaborativa en la actividad de reflexión			
2	Actitud Contextualización	El aprendiz participó de manera activa y colaborativa en la actividad de contextualización			
3	Actitud Apropiación	El aprendiz participó en las actividades didácticas para la apropiación del conocimiento			
4	Actitud Apropiación	El aprendiz se presenta puntual a las actividades de aprendizaje			
5	Actitud Apropiación	El aprendiz evidencia un buen porte del uniforme durante el proceso de aprendizaje			
6	Actitud Apropiación Técnico	El aprendiz realiza la elaboración del manual de usuario describiendo los conceptos básicos.			
8	Actitud Proyecto-transferencia	El aprendiz elabora el manual de usuario mediante la metodología scrum			
9	Actitud Portafolio de evidencias	El aprendiz registra las evidencias en su portafolio con base en las fechas e instrucciones dadas por el instructor			

RESULTADOS: Aprobado ☐ Deficiente ☐

OBSERVACIONES-RECOMENDACIONES

Ciudad y fecha de aplicación del instrumento:

_____ _____

Firma del Aprendiz Firma Instructor