



INSTITUTO TECNOLOGICO VIDA NUEVA

DOCUMENTACIÓN GENERAL

NOMBRE

STALIN ALEXIS CARRION JIMENEZ

PERIODO

ABRIL 2025 – SEPTIEMBRE 2025

Misión y Visión Institucional

Misión

El Tecnológico Universitario Vida Nueva (TUVN) forma profesionales: Técnicos, Tecnólogos, Tecnólogo Superior Universitario, Especialistas y Magísteres Tecnológicos con cualidades de liderazgo, capaces, competentes, emprendedores con vocación de servicio, profundo apego a principios éticos y espirituales, comprometidos con el desarrollo social, productivo y sustentable.

Visión

El Tecnológico Universitario Vida Nueva (TUVN) procura ser un centro de estudios con sólido prestigio, comprometido con oferta académica pertinente, para liderar procesos productivos y sociales a nivel nacional e internacional en diferentes modalidades de estudio, apoyados en sus principios y valores institucionales, enfocados en la investigación e innovación tecnológica y la contribución al desarrollo sustentable del país.

Planificación Académica General



FECHA	ACTIVIDAD
20/03/2025 - 26/03/2025	Jornada de planificación microcurricular.
14/04/2025	Inicio del periodo académico abril-septiembre 2025.
14/04/2025 - 05/05/2025	Proceso de selección de ayudantes de docencia.
01/05/2025	Jornada de crecimiento personal.
06/06/2025	Inauguración de campeonato interno de deportes.
17/07/2025	Jornada de Actualización Científica y Profesional TUVN Innova Talks.
12/09/2025	Finalización del periodo académico abril-septiembre 2025.

DÍAS LIBRES POR FERIADO O EVENTOS INSTITUCIONALES

18,19 y 20/04/2025	Semana santa.
01/05/2025	Jornada de Crecimiento Personal.
02/05/2025	Día del trabajo.
23 y 24/05/2025	Batalla de Pichincha.
06/06/2025	Inauguración de campeonato interno de deportes.
17 y 18/07/2025	Jornada de Actualización Científica y Profesional TUVN Innova Talks.
11/08/2025	Primer grito de la Independencia.

NIVEL	FECHA INICIO	CODIGO ASIGNATURA	ASIGNATURA	CRÉDITOS	HORARIO	AULA/TALLER
4	2025-04-14	25251781	PROGRAMACION WEB	4	10:00-12:00	LABE1
4	2025-04-14	25251761	BASE DE DATOS II	2	08:00-10:00	LABE1
4	2025-06-16	25251771	ANALISIS Y DISENO DE SOFTWARE	1	08:00-10:00	LABE1
4	2025-07-30	25251791	PROGRAMACION MOVIL	2	10:00-12:00	LABE1
4	2025-08-28	25251751	LEGISLACION LABORAL	1	08:00-10:00	F1

Horarios para evaluación de recuperación

HORARIOS PARA EVALUACIONES DE RECUPERACIÓN		
Modalidad (Jornada)	Horario de la asignatura reprobada	Horario para evaluación de recuperación
PRESENCIAL-HÍBRIDA (MATUTINA)	08:00-10:00 ; 07:00-10:00 ; 08:00-12:00	12:00-13:00
	10:00-12:00 ; 10:00-13:00 ; 12:00-14:00 ; 14:00-16:00	13:00-14:00
PRESENCIAL-HÍBRIDA (NOCTURNA)	18:00-20:00 ; 17:00-20:00 ; 18:00-22:00	19:00-20:00
	20:00-22:00 ; 14:00-16:00	21:00-22:00
PRESENCIAL (VESPERTINA)	13:00-15:00 ; 13:00-17:00	14:00-15:00
	15:00-17:00	16:00-17:00
SEMIPRESENCIAL-EN LÍNEA	08:00-12:00 o 07:00-12:00	17:00-18:00
	13:00-17:00 o 13:00-18:00	18:00-19:00

EL LUGAR DE LA EVALUACIÓN SERÁ NOTIFICADO POR COORD. DE CAMPUS POR MEDIO DE UN INFORMATIVO

Horario de acción tutorial

CARRERA	MODALIDAD PRESENCIAL /HÍBRIDA JORNADA MATUTINA/NOCTURNA	JUEVES
MECÁNICA AUTOMOTRIZ / ELECTROMECÁNICA AUTOMOTRIZ	12:10-13:10 16:50-17:50	Mg. Edwin Chiliquinga Mg. Jorge Ramos Mg. Jhon Jara Ing. Luis Iza Mg. Cristian Guachamin Ing. Denis Ugeño Mg. Edison Criollo
MECÁNICA INDUSTRIAL	12:10-13:10 16:50-17:50	Mg. Paúl Caza Mg. Víctor López Mg. Patricio Cruz Mg. Pamela Villareal Mg. Rodrigo Díaz
ELECTROMECÁNICA / AUTOMATIZACIÓN E INSTRUMENTACIÓN	12:10-13:10 16:50-17:50	Mg. Carlos Ruiz Mg. Darwin Tituaña Mg. Byron Machay Mg. Edwin Machay
DESARROLLO DE SOFTWARE	12:10-13:10 16:50-17:50	Mg. Darwin Arias Lcdo. Rubén Jaramillo Ing. Jorge Beltrán Ing. Luis Ulcuango Mg. Diego Camacho
CONTABILIDAD	12:10-13:10 16:50-17:50	Ing. Yesenia Alvarado Ing. Jessica Quinga
ADMINISTRACIÓN	12:10-13:10 16:50-17:50	Mg. Estefani Segura Mg. Paula Flores Eco. Carla Maldonado Mg. Jessica Quispe Mg. Cesar Tipán

SÁBADO

MODALIDAD SEMIPRESENCIAL

Mecánica Automotriz

Asignaturas del horario 08 - 12

07:00-07:50

Asignaturas del horario 13 - 18

18:10-19:00

MODALIDAD EN LÍNEA

Administración

Desarrollo de Software

Marketing Digital y Comercio Electrónico

Asignaturas del horario 08 - 12

07:00-07:50

Notas: no hay acción tutorial el día en que inicia una asignatura

Asignaturas del horario 13 - 17

17:10-18:00

Notas: no hay acción tutorial el día en que inicia una asignatura y tienen evaluación final

ENFERMERÍA

Para los docentes que se encuentren dictando clases en las jornadas matutina - vespertina vespertina - nocturna, el día acción tutorial se realizará los sábados en el horario de 08:00-09:00, de forma virtual.

Nota: Este horario se habilita durante el tiempo que este dictando asignaturas en las jornadas independientemente del horario. El control se realizará de acuerdo a los parámetros la modalidad En Línea.

EDUCACIÓN BÁSICA / DOCENCIA E INNOVACIÓN EDUCATIVA	12:10-13:10 16:50-17:50	Mg. Daysi Ainoca Mg. Maritza Quinzo Mg. Natalia Contero Mg. Diana Tonato Mg. Carolina Basantes Mg. Dario Guamán
TURISMO	12:10-13:10 16:50-17:50	Mg. Geovanny Cujano Ing. Estefania Segovia
ENFERMERÍA	12:10-13:10 16:50-17:50	Lcda. María José Villagrán Obst. Edgar Muñoz Lcda. Carla Becerra Lcda. Joselyn Pilicita Obst. Karla Bahamonde Lcda. Daniela Romero Lcdo. Oswaldo Copara Lcdo. Ricardo Toalombo
CENTRO DE IDIOMAS	16:50-17:50	Lcdo. Ricardo Quishpe Lcda. Katherine Sangoluisa Lcdo. Jorge Gavilanes Lcda. Jessica Toalombo Lcda. Mayra Arcos Lcda. Cristina Guanopatín



INSTITUTO TECNOLOGICO VIDA NUEVA

PROGRAMACIÓN WEB

DOCENTE

ING. BELTRAN ORTEGA JORGE DANIEL

NOMBRE

STALIN ALEXIS CARRION JIMENEZ

PERIODO

ABRIL 2025 - SEPTIEMBRE 2025

Syllabus



PROGRAMA DE ESTUDIO DE LA ASIGNATURA (Versión 2)

1. DATOS INFORMATIVOS

1.1.	Carrera:	Tecnología Superior en Desarrollo de Software		
1.2.	Modalidad:	Presencial		
1.3.	Nombre de la asignatura:	Programacion Web	DSUP12	
1.4.	Nivel:	Cuarto		
1.5.	Unidad de organización curricular:	Profesional		
1.6.	Número total de horas de la asignatura / créditos:	192 horas / 4 créditos		
1.7.	Número de horas del componente de aprendizaje en contacto con el docente:			64
1.8.	Número de horas del componente de aprendizaje práctico-experimental:			64
1.9.	Número de horas del componente de aprendizaje autónomo:			64
1.10.	Prerrequisito:	Asignatura:	Herramienta de diseño web	Código: DSUP09
1.11.	Correquisito:	Asignatura:	Ninguna	Código: Ninguna
1.12.	Línea(s) de vinculación con la sociedad y servicio a la comunidad relacionadas:	Programas y proyectos para el desarrollo educativo. Aplicación de conocimientos prácticos en la implementación de soluciones de hardware y software. Asesoramiento y capacitación en fundamentos básicos relacionados al campo Informático. Aplicación de conocimiento en el desarrollo y publicación de páginas web.		
1.13.	Línea(s) de investigación relacionadas:	Desarrollo y programación de los sistemas de información para aplicaciones web, distribuidas y dispositivos móviles		



2. CARACTERIZACIÓN DE LA ASIGNATURA

2.1. OBJETIVO DE LA ASIGNATURA

Aplicar los objetos (window, navigator, history, location) que nos permitirán acceder a información del navegador, abrir ventanas, crear mensajes y otra serie de tareas aplicando el lenguaje JavaScript para comprender las propiedades y métodos de estos objetos permitiendo obtener habilidades procedimentales para implementar scripts efectivos que optimicen la experiencia del usuario y gestionen la navegación web, fomentando la experimentación y la innovación en la creación de soluciones eficientes para proyectos web.

2.2. RESULTADOS DE APRENDIZAJE DEL PERFIL DE EGRESO A LOS QUE CONTRIBUYE LA ASIGNATURA

- Desarrollar aplicaciones de software utilizando tecnología de plataformas open Source, en ambientes local y en la web.
- Conoce los modelos conceptuales y tecnológicos relacionados con las Bases de Datos, y los utiliza como herramienta en la construcción de software.
- Construir software a partir de especificaciones de diseño, haciendo uso de las mejores prácticas que aseguren la calidad del producto.

2.3. CONTRIBUCIÓN A LAS PRÁCTICAS PREPROFESIONALES

En la ejecución de las prácticas preprofesionales el estudiante:

- Analizará los requerimientos funcionales y no funcionales, para el desarrollo de sistemas web dinámicos.
- Identificará las especificaciones preliminares para la construcción de sistemas web dinámicos en función a los requerimientos funcionales y no funcionales de los usuarios.
- Diseñará una arquitectura robusta para la construcción de sistemas web dinámicos con conexión a bases de datos relacionales.
- Construirá aplicaciones web altamente eficientes para la optimización de procesos críticos en sistemas de gestión de información.
- Aplicará pruebas manuales y automáticas en los sistemas web dinámicos para garantizar la calidad de software.
- Creará un plan de actualización y mejora continua de sistemas web para el desarrollo de nuevas funcionalidades a medida que el negocio lo vaya necesitando.



2.4. CONTRIBUCIÓN A LAS CAPACIDADES GENERALES PARA EL APRENDIZAJE Y FORMACIÓN CULTURAL

En cuanto a las capacidades generales para el aprendizaje y formación cultural, el estudiante será capaz de:

- Con la Filosofía aprenderán a cuestionar, analizar argumentos y desarrollar habilidades de pensamiento crítico que son aplicables al proceso de resolución de problemas técnicos en la programación web.
- Con la Psicología Organizacional proporciona conocimientos sobre dinámicas de grupo, liderazgo y colaboración efectiva, que son esenciales para el trabajo en equipo en proyectos de desarrollo de software.
- La Historia de la Ciencia y la Tecnología ayudaría a entender la evolución y el impacto de la tecnología, fomentando una mentalidad de aprendizaje continuo y adaptabilidad a nuevos avances tecnológicos.
- La importancia de la ética en el desarrollo de software, enseñando a los estudiantes a considerar el impacto de sus aplicaciones en la sociedad y a actuar con responsabilidad profesional. Esto contribuye a su formación integral como individuos conscientes y éticos.
- Al trabajar con tecnologías web que se utilizan en todo el mundo, los estudiantes adquieren una mayor apreciación de la diversidad cultural y los desafíos de la globalización. Aprenderán a diseñar aplicaciones inclusivas que consideren las necesidades de usuarios de diferentes contextos culturales y geográficos, promoviendo así una perspectiva global en su formación.



3. UNIDADES DIDÁCTICAS

NÚMERO DE LA UNIDAD:	Uno		
NOMBRE DE LA UNIDAD:	Introducción a JavaScript		
OBJETIVO DE LA UNIDAD:	Implementar el uso de eventos y validaciones mediante JavaScript para enriquecer la interactividad y dinamismo de estos elementos en el código de la página web con la finalidad de fomentar una actitud proactiva y de mejora continua, incentivando a los estudiantes a explorar nuevas formas de utilizar eventos y validaciones para mejorar la experiencia del usuario y la funcionalidad de las páginas web que desarrollen.		
ORGANIZACIÓN DEL APRENDIZAJE			
CONTENIDO MÍNIMO	APRENDIZAJE EN CONTACTO CON EL DOCENTE	APRENDIZAJE PRÁCTICO EXPERIMENTAL	APRENDIZAJE AUTÓNOMO
1. JavaScript.	Horas	Horas	Horas
1.1. Manejo de eventos. 1.2. Validación de formularios con JavaScript. 1.3. Calendarios con JavaScript. 1.4. JavaScript y HTML5.	8	2	4
ACTIVIDADES APRENDIZAJE	DE	Actividades de adquisición: Clases de programación en JavaScript. Actividades de colaboración: Desarrollar un formulario web de productos y clientes, generar el código en la IA generativa de ChatGPT, y comprobar su funcionamiento.	Actividades prácticas: Desarrollo de la guía práctica: "Algoritmos y programas (Programación con JavaScript)." Nombre de la guía: Algoritmos y programas (Programación con JavaScript) Objetivo: Crear estructuras HTML para el modelo de objetos del documento. Desarrollar métodos JavaScript para la funcionalidad de la interfaz gráfica. Implementar hojas de estilo en cascada para el diseño de la interfaz gráfica. Sustento:



		<p>La práctica propuesta posibilita que el estudiante desarrolle hojas de estilo en cascada para el diseño de la interfaz gráfica</p>	
RECURSOS DIDÁCTICOS	Proyector Parlantes Pc Roll up Navegadores SOFTWARES Microsoft Visual Studio Code	Pc Navegadores SOFTWARES Microsoft Visual Studio Code	Fotocopias Diapositivas Fotografías Infografías SOFTWARES Microsoft Visual Studio Code
ESCENARIOS DE APRENDIZAJE	Aula Laboratorio de computo	Aula	Campus virtual
CRITERIOS DE EVALUACIÓN	Indicador: Desarrolla aplicaciones formularias de acuerdo a los tipos de datos almacenado (Resolución de problemas)	Indicador: Aplica los conocimientos para el diseño de formularios web (Lista de cotejo)	Indicador: Aplica javascript con html y css en el desarrollo front end (Demostraciones)



NÚMERO DE LA UNIDAD:	Dos		
NOMBRE DE LA UNIDAD:	Manejo de ficheros y Ajax		
OBJETIVO DE LA UNIDAD:	Utilizar Ajax permitiendo la manipulación eficiente de datos entre la interfaz de usuario y el servidor web entender los principios y funcionamiento, garantizando una comunicación fluida, eficiente entre el cliente y el servidor con la finalidad de buscar nuevas formas de optimizar procesos en las aplicación web		
ORGANIZACIÓN DEL APRENDIZAJE			
CONTENIDO MÍNIMO	APRENDIZAJE EN CONTACTO CON EL DOCENTE	APRENDIZAJE PRÁCTICO EXPERIMENTAL	APRENDIZAJE AUTÓNOMO
1. Gestión de ficheros. 1.1. Leer archivos. 1.2. Escribir archivos. 1.3. Función file(). 1.4. Función file_put_contents(). 1.5. Utilidades del manejo de archivos con PHP. 2. AJAX 2.1. Ventajas de AJAX con PHP. 2.2. Manejo de formularios con AJAX.	Horas	Horas	Horas
	14	6	7
ACTIVIDADES APRENDIZAJE	DE	Actividades de adquisición: Clases de Gestión de ficheros. Clases de la sintaxis de AJAX. Clases de la sintaxis de AJAX para cargar y mostrar el contenido de archivos. Actividades de colaboración: Desarrollar una aplicación web para cargar y guardar el contenido en una base de datos de la tabla producto y cliente Desarrollar una aplicación que implemente métodos de seguridad para la carga de archivos. Desarrollar una aplicación web para cargar y guardar	Actividades prácticas: Desarrollo un programa web que permita cargar información de páginas externas mediante AJAX. Desarrollo de la guía práctica: "Algoritmos y programas (Formularios Ajax)." Nombre de la guía: Algoritmos y programas (Formularios Ajax) Objetivo: Crear estructuras HTML para el modelo de objetos del documento. Desarrollar métodos JavaScript para la funcionalidad de la interfaz gráfica. Implementar hojas de estilo en cascada para el
		Actividades de discusión: Discutir a través de un foro individual en la plataforma individual MOODLE el uso de input file, generar la información respectiva en la IA de ChatGPT y plantear en el foro su criterio con respecto a la lectura.	Actividades de producción: Evaluación de los contenidos de la unidad 1 Elaboración de un reporte sobre la carga información de páginas externas mediante AJAX. Elaboración del informe de la práctica "Algoritmos y programas (Programación con JavaScript)."



	el contenido en una base de datos.	diseño de la interfaz gráfica. Sustento: La práctica propuesta posibilita que el estudiante desarrolle hojas de estilo en cascada para el diseño de la interfaz gráfica	
RECURSOS DIDÁCTICOS	Proyector Parlantes Pc Roll up Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code	Pc Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code	Fotocopias Diapositivas Fotografías Infografías SOFTWARES Xampp MySQL Microsoft Visual Studio Code
ESCENARIOS APRENDIZAJE	DE Aula Laboratorio de computo	Aula	Campus virtual
CRITERIOS DE EVALUACIÓN	Indicador: Desarrolla aplicaciones basadas en Ajax con el uso de ficheros (Resolución de problemas)	Indicador: Aplica los conocimientos para un correcto para la funcionalidad del sistema por medio de ajax (Lista de cotejo)	Indicador: Realiza ejercicios que permitan leer y escribir sobre archivos (Demostraciones)



NÚMERO DE LA UNIDAD:	Tres		
NOMBRE DE LA UNIDAD:	Manejo de archivos con PHP		
OBJETIVO DE LA UNIDAD:	Dominar el lenguaje de programación PHP, la gestión de base de datos MySQL y el uso de JavaScript en el contexto de aplicaciones web, aplicando el modelo cliente-servidor, así como su interacción en el entorno web		
ORGANIZACIÓN DEL APRENDIZAJE			
CONTENIDO MÍNIMO	APRENDIZAJE EN CONTACTO CON EL DOCENTE	APRENDIZAJE PRÁCTICO EXPERIMENTAL	APRENDIZAJE AUTÓNOMO
1. Gestión de ficheros	Horas	Horas	Horas
1.1. Leer archivos. 1.2. Escribir archivos. 1.3. Función file (). 1.4. Función file_put_contents (). 1.5. Utilidades del manejo de archivos con PHP. 1.6. Manejo de archivos XML.	8	2	5
ACTIVIDADES DE APRENDIZAJE	DE	Actividades prácticas: Desarrollo de la guía práctica: "Algoritmos y programas (Manejo de archivos)." Nombre de la guía: Algoritmos y programas (Manejo de archivos) Objetivo: Desarrollar un sistema que permita leer, escribir archivos con PHP. Utilizar las etiquetas HTML para el manejo de archivos. Programar en PHP los métodos para el procesamiento de archivos de texto. Sustento: La práctica propuesta posibilita que el estudiante aplique distintos manejos de archivos	Actividades de discusión: Discutir a través de un foro individual sobre el uso de la función file_put_contents(). generar la información respectiva en la IA de ChatGPT y plantear en el foro su criterio con respecto a la lectura Actividades de producción: Evaluación de los contenidos de la unidad 2 Elaboración del informe de la práctica "Algoritmos y programas (Manejo de archivos)."



RECURSOS DIDÁCTICOS	Proyector Parlantes Pc Roll up Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code	Pc Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code	Fotocopias Diapositivas Fotografías Infografías SOFTWARES Xampp MySQL Microsoft Visual Studio Code
ESCENARIOS APRENDIZAJE	DE Aula Laboratorio de computo	Aula	Campus virtual
CRITERIOS DE EVALUACIÓN	Indicador: Desarrolla aplicaciones basadas en PHP con el uso de ficheros (Resolución de problemas)	Indicador: Aplica los conocimientos para un correcto para la gestión de archivos (Lista de cotejo)	Indicador: Explicación de actividad/información. (Demostraciones)



NÚMERO DE LA UNIDAD:	Cuatro			
NOMBRE DE LA UNIDAD:	Cookies y sesiones.			
OBJETIVO DE LA UNIDAD:	Implementar de manera efectiva el uso de funciones, separando la lógica de programación del sistema basado en la web de las interfaces HTML, siguiendo la arquitectura cliente-servidor			
ORGANIZACIÓN DEL APRENDIZAJE				
CONTENIDO MÍNIMO	APRENDIZAJE EN CONTACTO CON EL DOCENTE	APRENDIZAJE PRÁCTICO EXPERIMENTAL	APRENDIZAJE AUTÓNOMO	
1. Declaración de cookies. 2. Sesiones 2.1. Declaración de sesiones. 2.2. Variables de sesión. 2.3. Sistema de login. Sistema de carro de compras	Horas 25	Horas 15	Horas 17	
ACTIVIDADES APRENDIZAJE	DE	Actividades de adquisición: Clases de las cookies, sintaxis y definición. Clases de sesiones y sintaxis. Clases de sesiones y estructuras de control. Clases sobre construcción la capa de presentación del sistema de carro de compras. Clases sobre construcción de la capa de negocio del sistema de carro de compras. Actividades de colaboración: Construcción de un mapa mental sobre la declaración de cookies. Link: https://www.mindmeister.com/es Construcción de un mapa mental sobre el manejo de sesiones en PHP. Link: https://www.mindmeister.com/es Elaborar un proyecto con variables de sesiones en la plataforma GitHub. Link: https://github.com/ Elaborar un carrito de compras y subir en la plataforma GitHub. Link: https://github.com/	Actividades prácticas: Desarrollo de un login con sesiones. Desarrollo de la guía práctica: "Algoritmos y programas (Sistema de compras)." Nombre de la guía: Algoritmos y programas (Sistema de ingresos). Objetivo: Desarrollar un sistema de login para gestión de seguridad de un sistema web Crear una interfaz intuitiva para el login. Validar el ingreso de usuarios mediante con el uso de PHP. Sustento: La práctica propuesta posibilita que el estudiante aplique seguridades a nivel de acceso a la información Nombre de la guía:	Actividades de discusión: Discutir a través de un foro en la plataforma individual MOODLE sobre la seguridad para cookies, la información respectiva en la IA de ChatGPT y plantear en el foro su criterio con respecto a la lectura. Actividades de investigación: Analizar la información sobre las declaraciones en cookies en publicaciones y/o buscadores académicos Analizar la información sobre sesiones en PHP en publicaciones y/o buscadores académicos Actividades de producción: Elaboración de un reporte sobre la creación de un login con sesiones. Elaboración del informe de la práctica "Algoritmos y programas (Sistema de ingresos)."



	Realizar aportaciones en la capa de acceso a datos al desarrollo del carrito de compras en GitHub. Link: https://github.com/ Realizar aportaciones en la capa de interfaz en el desarrollo del carrito de compras en GitHub. Link: https://github.com/	"Algoritmos y programas (Sistema de compras) Objetivo: Desarrollar un sistema de carro de compras con PHP Mostrar una lista de productos disponibles para la compra. Implementar hojas de estilo en cascada para el diseño de la interfaz gráfica. Sustento: La práctica propuesta posibilita que el estudiante desarrolle sistemas enfocados a la realidad empresarial	
RECURSOS DIDÁCTICOS	Proyector Parlantes Pc Roll up Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code	Pc Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code	Fotocopias Diapositivas Fotografías Infografías SOFTWARES Xampp MySQL Microsoft Visual Studio Code
ESCENARIOS APRENDIZAJE	DE Aula Laboratorio de computo	Aula	Campus virtual
CRITERIOS EVALUACIÓN	DE Indicador: Desarrolla aplicaciones web utilizando variables de sesión y cookies (Resolución de problemas)	Indicador: Aplica los conocimientos para un correcto uso de variables de sesión y cookies (Lista de cotejo)	Indicador: Explicación de actividad/información. (Demostraciones)



NÚMERO DE LA UNIDAD:	Cinco				
NOMBRE DE LA UNIDAD:	Web Service				
OBJETIVO DE LA UNIDAD:	Dominar el uso de Web Services en el desarrollo de aplicaciones web, comprendiendo su funcionamiento para la comunicación entre sistemas distribuidos cumpliendo los estándares relacionados con Web Services, como SOAP y REST.				
ORGANIZACIÓN DEL APRENDIZAJE					
CONTENIDO MÍNIMO	APRENDIZAJE EN CONTACTO CON EL DOCENTE	APRENDIZAJE EN CONTACTO CON EL DOCENTE	APRENDIZAJE EN CONTACTO CON EL DOCENTE		
	Horas	Horas	Horas		
1. Definición de un Web Service 1.1. Soluciones de Web Service con PHP. 2. Creación de un Web Service. 2.1. Creación de un cliente para el Web Service. 2.2. Sistema de control estudiantil PHP.	31	9	39		
ACTIVIDADES APRENDIZAJE	DE	Actividades de adquisición: Clases de la definición de un web services Clases de la estructura de un web services Clases sobre el uso y manejo de NuSOAP Clases sobre la gestión del web services	Actividades de adquisición: Clases de la definición de un web services Clases de la estructura de un web services Clases sobre el uso y manejo de NuSOAP Clases sobre la gestión del web services	Actividades de adquisición: Clases de la definición de un web services Clases de la estructura de un web services Clases sobre el uso y manejo de NuSOAP Clases sobre la gestión del web services	Actividades de adquisición: Clases de la definición de un web services Clases de la estructura de un web services Clases sobre el uso y manejo de NuSOAP Clases sobre la gestión del web services
	DE	Clases sobre la manipulación de contenidos dentro del web services Aplicación de la evaluación final de la asignatura Socialización de las notas parciales y final de la asignatura conjuntamente con la retroalimentación de su contenido.	Clases sobre la manipulación de contenidos dentro del web services Aplicación de la evaluación final de la asignatura Socialización de las notas parciales y final de la asignatura conjuntamente con la retroalimentación de su contenido.	Clases sobre la manipulación de contenidos dentro del web services Aplicación de la evaluación final de la asignatura Socialización de las notas parciales y final de la asignatura conjuntamente con la retroalimentación de su contenido.	Clases sobre la manipulación de contenidos dentro del web services Aplicación de la evaluación final de la asignatura Socialización de las notas parciales y final de la asignatura conjuntamente con la retroalimentación de su contenido.
		Actividades de colaboración:			Actividades de colaboración:



	Construcción de un mapa mental sobre beneficios y utilidades de web services Construcción de un mapa mental sobre beneficios y utilidades de web services Sustentación del trabajo final por equipos	Actividades de colaboración: Construcción de un mapa mental sobre beneficios y utilidades de web services Construcción de un mapa mental sobre beneficios y utilidades de web services Sustentación del trabajo final por equipos	Construcción de un mapa mental sobre beneficios y utilidades de web services Construcción de un mapa mental sobre beneficios y utilidades de web services Sustentación del trabajo final por equipos
RECURSOS DIDÁCTICOS	Proyector Parlantes Pc Roll up Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code SlimFramework composer	Pc Navegadores SOFTWARES Xampp MySQL Microsoft Visual Studio Code SlimFramework composer	Fotocopias Diapositivas Fotografías Infografías SOFTWARES Xampp MySQL Microsoft Visual Studio Code SlimFramework composer
ESCENARIOS APRENDIZAJE	DE Aula Laboratorio de computo	Aula Laboratorio de computo	Campus virtual
CRITERIOS DE EVALUACIÓN	Indicador: Desarrolla aplicaciones web utilizando web services (Resolución de problemas)	Indicador: Aplica los conocimientos para un correcto uso de web services (Lista de cotejo)	Indicador: Explicación de actividad/información. (Demostraciones)



4. METODOLOGÍAS DE ENSEÑANZA

METODOLOGÍA Para el desarrollo de las clases se pueden usar las siguientes metodologías	MÉTODO	ESTRATÉGIA
	Controlado por el docente	Deductivo Enunciación Comprobación Aplicación Analógico-comparativo Observación Comparación Conclusiones
	Controlado por el estudiante	Método activo o clases prácticas Enunciación Aclaración Resolución Comisión Planteamiento del problema Análisis Conclusiones
	Controlado por el docente y el estudiante	Aprendizaje basado en proyectos Diseño Planificación Ejecución Evaluación Observación Construcción de supuestos preliminares Búsqueda de información Establecimiento de semejanzas y diferencias Elaboración de descripciones sobre lo observado
SUSTENTO:		
Controlado por el docente:		
<ul style="list-style-type: none">Deductivo: Este enfoque fomenta la capacidad de los estudiantes para aplicar principios generales a casos particulares, desarrollando habilidades de razonamiento lógico y deductivo. Por ejemplo, al enseñar sobre funciones en PHP, el docente puede presentar primero la teoría general de las funciones y luego mostrar cómo se implementan en diferentes escenarios web, como formularios de contacto o sistemas de autenticación.Analógico-comparativo: Las analogías y comparaciones crean conexiones más fuertes entre los conceptos nuevos y conocidos, mejorando la retención de información. Un docente puede comparar la estructura de un documento HTML con la de un documento en Word, explicando cómo las etiquetas HTML funcionan de manera similar a los estilos y secciones de un documento de texto tradicional.		



Controlado por el estudiante:

- **Enunciación:** Fomenta la capacidad de los estudiantes para expresar ideas y conceptos de manera clara y precisa, una habilidad crucial tanto en el ámbito académico como profesional. En programación web, los estudiantes pueden ser incentivados a explicar en voz alta el propósito y funcionamiento de su código, ya sea en discusiones grupales o mediante presentaciones, fortaleciendo así su comprensión y habilidades comunicativas.
- **Estudio de caso:** Ayuda a los estudiantes a entender cómo se aplican los conceptos en contextos específicos, preparando mejor para situaciones profesionales. Analizando casos reales de sitios web, como el diseño de una tienda en línea o una plataforma de blog, los estudiantes pueden ver cómo se aplican HTML, CSS, PHP y MySQL para resolver problemas concretos.

Controlado por el docente y el estudiante:

- **Aprendizaje basado en proyectos:** Permite a los estudiantes y docentes trabajar juntos, garantizando que se cumplan los objetivos educativos mientras se promueve la autonomía estudiantil. Por ejemplo, desarrollar un proyecto completo como un sistema de gestión de contenido (CMS) permite a los estudiantes aplicar lo aprendido en clase en un proyecto real, con el docente guiando y supervisando el proceso.
- **Observación:** Involucra la recopilación de datos y la percepción directa de fenómenos o experimentos. En programación web, esto puede traducirse en revisar y analizar el comportamiento de diferentes sitios web, identificando buenas prácticas y errores comunes. Esta observación activa ayuda a los estudiantes a aprender de ejemplos reales y aplicar esos aprendizajes en sus propios proyectos.

5. BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA:	
1	Heurtel, O. (2023). <i>Php y Mysql. Domine el Desarrollo de un Sitio Web Dinamico e Interactivo</i> (5 ^a ed.). Eni. (005.8111 - Libro Físico)
2	Christophe, A. (2021). <i>Html5 y Css3. Domine los Estandares de la Creacion de Sitos Wed</i> (2 ^a ed.). Eni. (005.8067 - Libro Físico)

SUSTENTO:	Científico - Técnico
	Estos libros proporcionan acceso a conocimientos especializados y actualizados sobre tecnologías web fundamentales. El libro de Heurtel Oliver abarca PHP y MySQL, ofreciendo una comprensión profunda del desarrollo de sitios web dinámicos, mientras que el de Aubry Christopher se centra en HTML5 y CSS3, esenciales para la creación de interfaces web modernas. Estos textos han sido revisados por expertos, asegurando la precisión y relevancia de la información presentada. Además, cubren metodologías y prácticas recomendadas en el campo de la programación web, proporcionando una base sólida para la investigación y la aplicación práctica de estos lenguajes.



	<p>Dicactico</p> <p>Desde la perspectiva didáctica, estos libros están diseñados de manera que facilitan el aprendizaje progresivo y autónomo. Están organizados temáticamente, lo que permite a los estudiantes avanzar desde conceptos básicos hasta avanzados de manera estructurada. Incluyen ejemplos prácticos, ejercicios, y proyectos que ayudan a consolidar los conocimientos adquiridos. Además, el uso de diagramas, capturas de pantalla y códigos comentados en los libros facilita la comprensión visual y práctica de los conceptos. La disponibilidad de estos libros en formato físico permite a los estudiantes acceder a ellos en cualquier momento, proporcionando una fuente de referencia constante y confiable para resolver dudas y profundizar en los temas según sus necesidades individuales.</p>
BIBLIOGRAFÍA DE CONSULTA:	
1	Pavon Puertas, J. (2007). <i>Creacion de un Portal con Php y Mysql</i> (3 ^a ed.). Alfaomega. (007.5299 - Libro Digital)

6. SISTEMA DE EVALUACIÓN GENERAL

OBJETIVO DE LA UNIDAD	CRITERIO DE EVALUACIÓN	ACTIVIDAD	PRODUCTO
Implementar el uso de eventos y validaciones mediante JavaScript para enriquecer la interactividad y dinamismo de estos elementos en el código de la página web con la finalidad de fomentar una actitud proactiva y de mejora continua, incentivando a los estudiantes a explorar nuevas formas de utilizar eventos y validaciones para mejorar la experiencia del usuario y la funcionalidad de las páginas web que desarrollean.	Desarrolla aplicaciones formularias de acuerdo a los tipos de datos almacenado	Diseñar un formulario web con validaciones JavaScript para cada campo.	Resolución de problemas
	Aplica los conocimientos para el diseño de formularios web	Crear una lista de verificación para evaluar el formulario.	Lista de cotejo
	Explicación de actividad/información.	Demostrar cómo implementar eventos y validaciones en JavaScript.	Demostraciones.
Utilizar Ajax permitiendo la manipulación eficiente de datos entre la interfaz de usuario y el servidor web entender los principios y funcionamiento, garantizando una comunicación fluida, eficiente entre el cliente y el servidor con la finalidad de buscar nuevas formas de	Desarrolla aplicaciones basadas en Ajax con el uso de ficheros	Desarrollar una aplicación web que use Ajax para CRUD con datos en un archivo JSON.	Resolución de problemas
	Aplica los conocimientos para un correcto para la funcionalidad del sistema por medio de ajax	Crear una lista de verificación para evaluar la implementación de Ajax.	Lista de cotejo.



optimizar procesos en las aplicación web	Explicación de actividad/información.	Demostrar cómo implementar Ajax para la manipulación de datos.	Demostraciones
Dominar el lenguaje de programación PHP, la gestión de base de datos MySQL y el uso de JavaScript en el contexto de aplicaciones web, aplicando el modelo cliente-servidor, así como su interacción en el entorno web	Desarrolla aplicaciones basadas en PHP con el uso de ficheros	Crear una aplicación web que maneje archivos (lectura, escritura, actualización, eliminación) utilizando PHP y MySQL.	Resolución de problemas
	Aplica los conocimientos para un correcto para la gestión de archivos	Crear una lista de verificación para evaluar la gestión de archivos en PHP.	Lista de cotejo
	Explicación de actividad/información.	Demostrar cómo usar PHP y MySQL para gestionar archivos en una aplicación web.	Demostraciones
Implementar de manera efectiva el uso de funciones, separando la lógica de programación del sistema basado en la web de las interfaces HTML, siguiendo la arquitectura cliente-servidor .	Desarrolla aplicaciones web utilizando variables de sesión y cookies	Crear una aplicación web que utilice variables de sesión y cookies para gestionar la autenticación de usuarios y preferencias personalizadas.	Resolución de problemas.
	Aplica los conocimientos para un correcto uso de variables de sesión y cookies	Crear una lista de verificación para evaluar el uso de sesiones y cookies en la aplicación web.	Lista de cotejo.
	Explicación de actividad/información.	Demostrar cómo implementar variables de sesión y cookies en una aplicación web.	Demostraciones



Dominar el uso de Web Services en el desarrollo de aplicaciones web, comprendiendo su funcionamiento para la comunicación entre sistemas distribuidos cumpliendo los estándares relacionados con Web Services, como SOAP y REST.	Desarrolla aplicaciones web utilizando web services	Crear una aplicación web que consuma un Web Service (SOAP o REST) para obtener y mostrar datos dinámicos.	Resolución de problemas.
	Aplica los conocimientos para un correcto uso de web services	Crear una lista de verificación para evaluar la integración y el uso de Web Services en la aplicación web.	Lista de cotejo
	Explicación de actividad/información.	Demostrar cómo implementar y consumir Web Services (SOAP/REST) en una aplicación web.	Demostraciones.



7. REVISIÓN Y APROBACIÓN

ELABORADO	REVISADO	APROBADO
<p>COMISIÓN DE LA CARRERA</p> <p>FECHA: 02/10/2023</p> <p><i>Coordinador de carrera: Sede Matriz</i></p> <p>_____ Ing. Arias Martinez Darwin René Mg.</p> <p>Coordinador de carrera Campus Norte: _____ Ing. Toapanta Kevin</p> <p><i>Docentes de la carrera</i></p>	<p>COORDINACIÓN ACADÉMICA</p> <p>FECHA: 06/10/2023</p> <p>_____ Coordinador Académico Sede Matriz</p> <p>_____ Coordinador Académico Campus Norte</p>	<p>VICERRECTORADO RECTORADO</p> <p>FECHA: 10/10/2023</p> <p>_____ Vicerrectora General</p> <p>_____ Rector</p>

8. HISTORIAL DE ACTUALIZACIONES

VERSIÓN	FECHA DE REVISIÓN	FECHA DE ACTUALIZACIÓN	ACTUALIZACIÓN	SUSTENTO DE LA ACTUALIZACIÓN
Versión 1	-----	01-03-2023	Cambio de Formato	Modelo de evaluación externa 2024 con fines de acreditación para los Institutos Superiores Técnicos y Tecnológicos.
Versión 2	02-10-2023	02-10-2023	Bibliografía básica	Informe de seguimiento, control y evaluación del proceso docente.



9. TRABAJO FINAL DE ASIGNATURA

	INSTITUTO SUPERIOR UNIVERSITARIO VIDA NUEVA
--	--

TRABAJO FINAL DE ASIGNATURA	
DATOS INFORMATIVOS	
TEMA:	Algoritmos y programas (Sistema de Inventario)
OBJETIVO GENERAL:	Desarrollar un sistema de control Inventario para la administración de una Ferretería utilizando web services
ESTRATEGIA/TÉCNICA:	Aprendizaje Basado en Proyectos
PASOS / ACTIVIDADES:	<ul style="list-style-type: none">● Diseño Establezca los requerimientos funcionales y no funcionales para la implementación del sistema. Determine la cantidad de usuarios que van a utilizar el sistema. Establezca un diseño de base de datos que soporte los requerimientos funcionales del sistema. Modelar la base de datos.● Planificación Construya la arquitectura que será utilizada para la construcción del software.● Ejecución Realice la codificación de las interfaces en lenguaje HTML, CSS y JavaScript. Desarrolle el código del servidor en lenguaje PHP para la interacción entre las interfaces web y la base de datos● Evaluación Realice pruebas de calidad del software y de cumplimiento de requerimientos funcionales y no funcionales
BIBLIOGRAFÍA:	Heurtel Oliver (2023). <i>Php y mysql. Domine el desarrollo de un sitio web dinamico e interactivo</i> . España: ENI (005.8111-Físico) Aubry Christopher (2021) <i>Html5 y css3. Domine los estandares de la creacion de sitios web</i> España ENI (005.8067-Físico)



**INSTITUTO SUPERIOR UNIVERSITARIO
VIDA NUEVA**

RÚBRICA DEL TRABAJO FINAL DE ASIGNATURA

INFORME/0.5:

CONSIDERACIONES	0,10 MUY BUENO	0, 05 BUENO	0, 03 REGULAR
Ortografía	No existen errores ortográficos en el informe	Se detecta un error ortográfico en el informe	Se detecta 2 o 3 errores ortográficos en el informe
Contenido	Toda la información en el informe es coherente con el tema planteado	La mayor parte de los contenidos en el informe es coherente con el tema planteado.	Los contenidos en el informe no son nada coherentes con el tema planteado.
Formato	Cumple exitosamente con el formato establecido dentro del informe.	Cumple solo con una parte del formato establecido en el informe.	No contiene ningún parámetro dentro del formato establecido en el informe.
Método y técnicas de presentación	La ejecución de la aplicación contiene todos los parámetros requeridos.	La ejecución de la aplicación no contiene todos los parámetros requeridos.	La ejecución de la aplicación no contiene ningún parámetro requerido.
Originalidad en la presentación	Muestra contenido original y nuevas técnicas de presentación para el proyecto.	Denota poca originalidad sin nuevas técnicas de presentación para el proyecto.	No muestra originalidad ni tampoco técnicas de presentación para el proyecto.

DESARROLLO-SEGUIMIENTO-SUSTENTACIÓN/0.5:

CONSIDERACIONES	0,10 MUY BUENO	0,05 BUENO	0,01 REGULAR
Arquitectura de la aplicación	La aplicación contiene una arquitectura perfecta	La aplicación contiene una arquitectura parcialmente estructurada	La aplicación no contiene una arquitectura definida
Diseño de la capa "Vista"	El diseño es coherente mediante el uso de la psicología de colores y la accesibilidad de la aplicación con el usuario.	El diseño es parcialmente correcto con el uso de la psicología de colores y la accesibilidad de la aplicación con el usuario.	El diseño es erróneo con el uso de la psicología de colores y la accesibilidad de la aplicación con el usuario.
Diseño y arquitectura de la base de datos	La aplicación contiene una arquitectura de base de datos perfecta	La aplicación contiene una arquitectura de base de datos parcialmente estructurada	La aplicación no contiene una arquitectura de base de datos.
Funcionalidad	La aplicación cumple con todos los requerimientos del sistema	A aplicación tiene un funcionamiento parcialmente funcional	No funciona el sistema o sus requerimientos.



Tecnológico Universitario
Vida Nueva



Página 22 de 24

Contestar preguntas	Demuestra un conocimiento completo del tema.	Demuestra un buen conocimiento del tema.	No parece conocer muy bien el tema.	
---------------------	--	--	-------------------------------------	--



10. ORGANIZACIÓN DEL APRENDIZAJE:

Descripción de la asignatura:	Desarrollar e implementar sistemas de información en ambientes web para la automatización de procesos y toma de decisiones, utilizando metodologías basadas en estándares internacionales y tecnologías emergentes																											
	<p>Gráfica del aprendizaje combinado</p> <table border="1"> <thead> <tr> <th>Método</th> <th>% Estimado</th> <th>% Planificado</th> </tr> </thead> <tbody> <tr> <td>Asíncrono</td> <td>36,84</td> <td>37,50</td> </tr> <tr> <td>Síncrono (cara a cara)</td> <td>63,16</td> <td>62,50</td> </tr> </tbody> </table> <p>■ Asíncrono ■ Síncrono (cara a cara)</p> <p>Tipos de aprendizaje estimados y planificados</p> <table border="1"> <thead> <tr> <th>Categoría</th> <th>Adquisición (%)</th> <th>Producción (%)</th> <th>Práctica (%)</th> <th>Discusión (%)</th> <th>Investigación (%)</th> </tr> </thead> <tbody> <tr> <td>Estimado (Azul)</td> <td>60</td> <td>40</td> <td>30</td> <td>20</td> <td>10</td> </tr> <tr> <td>Planificado (Naranja)</td> <td>40</td> <td>30</td> <td>20</td> <td>10</td> <td>10</td> </tr> </tbody> </table>	Método	% Estimado	% Planificado	Asíncrono	36,84	37,50	Síncrono (cara a cara)	63,16	62,50	Categoría	Adquisición (%)	Producción (%)	Práctica (%)	Discusión (%)	Investigación (%)	Estimado (Azul)	60	40	30	20	10	Planificado (Naranja)	40	30	20	10	10
Método	% Estimado	% Planificado																										
Asíncrono	36,84	37,50																										
Síncrono (cara a cara)	63,16	62,50																										
Categoría	Adquisición (%)	Producción (%)	Práctica (%)	Discusión (%)	Investigación (%)																							
Estimado (Azul)	60	40	30	20	10																							
Planificado (Naranja)	40	30	20	10	10																							



Plantillas ABC

Guías para prácticas



Tecnológico Universitario *Vida Nueva*



Práctica No. 1

Código: DSUP12-GP-01

Nombre: Stalin Carrión

Título: Algoritmos y programas (Programación con JavaScript).

Tiempo de duración: 120 minutos

Fundamentos:

JavaScript es un lenguaje de programación ampliamente utilizado en el desarrollo web debido a su capacidad para interactuar dinámicamente con el contenido de las páginas web. Entender los conceptos fundamentales de JavaScript es esencial para poder desarrollar aplicaciones web interactivas y dinámicas.

En JavaScript, las variables son contenedores para almacenar valores. Se pueden declarar utilizando las palabras clave "var", "let" y "const".

var: Permite declarar variables globales o locales en una función.

let: Introducido en ES6, permite declarar variables limitadas al ámbito de un bloque, declaración o expresión.

const: También introducido en ES6, declara variables cuyos valores son constantes y no pueden ser reasignados.

JavaScript cuenta con varios tipos de datos fundamentales: Números:

Enteros y de punto flotante.

Cadenas (Strings): Secuencias de caracteres encerradas entre comillas simples o dobles.

Booleanos: Valores lógicos true (verdadero) o false (falso).

Objetos: Estructuras de datos complejas que pueden contener datos y funcionalidad.

Arrays (Matrices): Estructuras de datos para almacenar múltiples elementos.

Funciones.



Tecnológico Universitario *Vida Nueva*



Las funciones en JavaScript son bloques de código reutilizables que se utilizan para realizar tareas específicas. Pueden tomar argumentos y devolver valores

Objetivos

- Crear estructuras HTML para el modelo de objetos del documento.
- Desarrollar métodos JavaScript para la funcionalidad de la interfaz gráfica.
- Implementar hojas de estilo en cascada para el diseño de la interfaz gráfica

Materiales y Herramientas

- Software (Sublime Text 3, Visual Code)
- Software (XAMPP, Laragon)
- Computador



Tecnológico Universitario *Vida Nueva*



Normas de Seguridad

Seguridad Física

Es imperativo mantener el área de trabajo ordenada en todo momento. Los cables deben estar adecuadamente gestionados y asegurados para evitar riesgos de tropiezos y otros accidentes. Se recomienda el uso de sujetacables y canaletas para mantener los cables organizados y fuera del paso.

El acceso al laboratorio debe estar restringido únicamente a estudiantes matriculados y el docente en los cursos correspondientes.

Los equipos y periféricos deben ser manipulados con sumo cuidado. Se debe verificar regularmente que todos los dispositivos estén correctamente conectados y en buen estado de funcionamiento. Se deben seguir las directrices del fabricante para el manejo, almacenamiento y mantenimiento de los equipos, para asegurar su funcionamiento óptimo y prolongar su vida útil.

Seguridad Informática

Todas las aplicaciones deben mantenerse actualizadas con los últimos parches de seguridad. Se debe instalar y mantener actualizado un software antivirus y un firewall en todos los equipos. Es esencial realizar análisis regulares del sistema para detectar y eliminar posibles amenazas de malware y proteger los datos e información del laboratorio.

Conducta Profesional y Ética

Se debe respetar estrictamente las licencias de software y las políticas de uso justo. Está prohibido el uso del laboratorio para actividades no autorizadas, incluyendo la descarga de software ilegal o cualquier otro uso indebido de los recursos del laboratorio.



Tecnológico Universitario **Vida Nueva**



Preparación Previa del Estudiante

Conocimiento Básico de Lógica y Algoritmos

Los estudiantes deben comprender los conceptos fundamentales de lógica, incluyendo operaciones lógicas (AND, OR, NOT), tablas de verdad y operadores condicionales. Este conocimiento es crucial para desarrollar habilidades en la creación de condiciones y la toma de decisiones en el código.

Es importante tener una comprensión preliminar de qué es un algoritmo y cómo se utiliza para resolver problemas de manera sistemática. Esto incluye el diseño y análisis de algoritmos básicos para la resolución de problemas computacionales.

Los estudiantes deben poseer habilidades en matemáticas básicas. Estos conocimientos son esenciales para la creación y comprensión de algoritmos y para la manipulación de datos en programación.

Lectura Previa

Antes de comenzar la asignatura, se recomienda leer material introductorio sobre conceptos clave de programación. Esto incluye variables, estructuras de control (como sentencias if y loops), funciones y estructuras de datos básicas (como listas y arrays).

La lectura de guías o tutoriales sobre la sintaxis básica de JavaScript es esencial. Esto incluye cómo declarar variables, utilizar operadores, controlar el flujo del programa y manipular datos. Recursos como la Documentación Oficial de JavaScript (MDN Web Docs) son un buen punto de partida.

Práctica en Escritura de Esquemas de Flujo

Los estudiantes deben practicar la creación de diagramas de flujo para representar algoritmos de manera visual. Esto ayuda a entender el flujo lógico de los programas y facilita la transición de algoritmos a código.



La práctica de escribir pseudocódigo para resolver problemas simples es crucial.

Permite a los estudiantes traducir ideas de solución a un formato estructurado que luego puede ser convertido en código JavaScript.

Realizar ejercicios que requieran razonamiento lógico y resolución de problemas, como rompecabezas y problemas matemáticos básicos, mejora las habilidades necesarias para la programación.

Revisión de Normas de Seguridad

Es esencial que los estudiantes revisen las normas básicas de seguridad informática, incluyendo la gestión segura de contraseñas, la importancia de las actualizaciones de software y la conciencia sobre amenazas comunes como malware y phishing.

Los estudiantes deben estar familiarizados con las normas de seguridad específicas del laboratorio de programación, incluyendo el uso adecuado del equipo, la política de acceso y conducta, y las medidas para proteger la información y los dispositivos.

Preparación de Herramientas y Materiales

Los estudiantes deben instalar un entorno de desarrollo para JavaScript. Se recomienda descargar e instalar Node.js para ejecutar JavaScript fuera del navegador y un IDE como Visual Studio Code.

Los estudiantes deben configurar su entorno de desarrollo, familiarizándose con su uso básico, incluyendo la escritura, ejecución y depuración de código JavaScript.

Preparar libros de texto recomendados, enlaces a recursos en línea, y apuntes de clase. Además, es útil tener a mano una lista de recursos adicionales como foros, tutoriales en video, y grupos de estudio en línea.



Procedimiento

1. Crear las estructuras para el modelo de objetos del documento HTML.

Figura 1

Calendario JS

May 2018						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

[Previous](#) [Next](#)

Jump To:

Nota. Desarrollo de calendario con JavaScript.

Figura 2

Estructura para modelo de objetos

```
Document
└── html
    ├── head
    │   ├── meta (charset)
    │   ├── meta (viewport)
    │   ├── title
    │   └── style
    └── body
        └── div (id="calendario")
            ├── header
            │   └── h2
            ├── div (class="dia encabezado") * 7
            └── div (class="dia") * número de días del mes
```

Nota. objetos con html.



2. Programar los métodos JavaScript para la creación y funcionalidad del calendario.

Crea un nuevo elemento HTML especificado por la etiqueta proporcionada.

Utilizado para generar dinámicamente los días del mes en el calendario.

Figura 3

Creación de nuevos elementos HTML

```
const celda = document.createElement('div');
celda.classList.add('dia');
```

Nota. Manejo desde la etiqueta proporcionada.

Devuelve una referencia al primer objeto con el valor de ID especificado.

Utilizado para seleccionar el contenedor del calendario donde se insertarán los días.

Figura 4

Document.getElementById

```
const calendario = document.getElementById('calendario');
```

Nota. Uso de valores por referencia.

Añade un nodo como el último hijo de un nodo padre. Utilizado para agregar los elementos de los días al contenedor del calendario.

Utilizado para seleccionar el contenedor del calendario donde se insertarán los días.

Figura 5

AppendChild

```
calendario.appendChild(celda);
```

Nota. Añade un nodo como el último hijo de un nodo padre.



3. Diseñar la interfaz gráfica mediante el desarrollo de hojas de estilo en cascada.

Para complementar la funcionalidad de un calendario en JavaScript con un diseño atractivo, puedes emplear CSS para estilizar la interfaz.

Figura 6

Uso de CSS

```
.calendario {  
    width: 350px;  
    background: #fff;  
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
    border-radius: 8px;  
    overflow: hidden;  
    font-size: 14px;  
}  
  
.calendario header {  
    background: #4caf50;  
    color: #fff;  
    text-align: center;  
    padding: 10px;  
}
```

Nota. En la imagen se muestra la implementación de CSS.

Para que esta funcionalidad sea correcta, se deberá contar con los selectores aplicados en el código HTML

Figura 7

Selectores con nombres en los id o las class en html

```
<body>  
    <div class="calendario-control">  
        <button id="btnPrev">Anterior</button>  
        <button id="btnNext">Siguiente</button>  
    </div>  
    <div id="calendario" class="calendario">  
        <header>  
            <h2>Julio 2024</h2>  
        </header>  
        <div class="dia encabezado">Lun</div>
```

Nota. En la imagen se muestra la implementación de CSS en html.



4. Utilizar Jquery para modelar la interfaz gráfica de usuario.

Figura 8

Llamado a la librería de JQuery

```
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.
    <title>Calendario con jQuery</title>
    <link rel="stylesheet" href="estilos-calendario.css">
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="calendario.js"></script>
</head>
<body>
```

Nota. En la imagen se muestra el uso de la librería externa de jquery.

5. En el contexto de la implementación de un calendario con jQuery, CSS se utiliza para definir el estilo y el diseño del calendario. A continuación, se detallan algunos selectores CSS utilizados en el archivo estilos- calendario.css:

Tabla 1

Detalle de selectores

SELECTORES CSS	
.class	Se usa para aplicar estilos a los días del calendario con .dia.
#id	Se aplica a los contenedores principales como #calendario-container y #dias.
:first-child	Puede usarse para aplicar estilos al primer día del calendario.
*	Puede emplearse para resetear márgenes o aplicar estilos globales.

Nota. En la tabla colocar los selectores empleados en el calendario.



Descripción de Resultados Obtenidos

Diseño de la estructura en html

Se creó una estructura básica que incluye un contenedor principal para el calendario (#calendario-container) y un espacio destinado para los días (#días). Esta organización facilita el manejo de estilos y la inserción dinámica de contenido desde JavaScript.



```
1 <!-- index.html -->
2 <!DOCTYPE html>
3 <html Lang="es">
4 <head>
5   <meta charset="UTF-8">
6   <title>Calendario JS</title>
7   <link rel="stylesheet" href="estilos-calendario.css">
8 </head>
9 <body>
10  <div id="calendario-container">
11    <h1>Calendario JS</h1>
12    <div id="navegacion">
13      <button id="anterior">Anterior</button>
14      <span id="mes-actual"></span>
15      <button id="siguiente">Siguiente</button>
16    </div>
17    <div id="dias"></div>
18  </div>
19  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
20  <script src="calendario.js"></script>
21 </body>
22 </html>
23
```

Ejemplo de Estilos con CSS en las clases

Se aplicaron estilos con la clase .día para definir la apariencia de cada día del calendario. Esto incluye fondo verde claro, bordes redondeados, espaciado interno y efectos visuales al pasar el cursor (:hover), logrando una interfaz limpia y moderna.



```
1 /* Estilos para el botón anterior y siguiente */
2 #anterior,
3 #siguiente {
4   background-color: #007bff;
5   color: white;
6   border: none;
7   padding: 10px 20px;
8   border-radius: 5px;
9   cursor: pointer;
10  font-size: 16px;
11  margin: 10px;
12  transition: background-color 0.3s ease;
13 }
14 #anterior:hover,
15 #siguiente:hover {
16   background-color: #0056b3;
17 }
```

Uso de Javascript

JavaScript se utilizó para generar dinámicamente los días del mes según el mes y año actual. Usando funciones como createElement y appendChild, se crearon nodos HTML que fueron insertados dentro del contenedor de días. También se añadió interactividad básica con jQuery, como alertar al seleccionar un día.



```
1 // calendario.js
2 let mesActual = new Date().getMonth();
3 let añoActual = new Date().getFullYear();
4
5 function crearCalendario(mes, año) {
6     const diaDelMes = new Date(año, mes + 1, 0).getDate();
7     const contenedor = document.getElementById('dias');
8     contenedor.innerHTML = ''; // Limpia el contenedor
9
10    const hoy = new Date();
11
12    for (let dia = 1; dia <= diaDelMes; dia++) {
13        const nuevoDia = document.createElement('div');
14        nuevoDia.className = 'dia';
15        nuevoDia.textContent = dia;
16
17        // Resaltar el día actual
18        if (dia === hoy.getDate() && mes === hoy.getMonth() && año === hoy.getFullYear()) {
19            nuevoDia.classList.add('today');
20        }
21
22        contenedor.appendChild(nuevoDia);
23    }
24 }
```

Resultado

El resultado es un calendario funcional y visualmente agradable, que se genera de forma automática y puede adaptarse a distintos meses y años. La combinación de HTML, CSS y JavaScript permite una interfaz gráfica dinámica, interactiva y personalizable.



1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Conclusiones

- La estructura HTML permite modelar el contenido de la página, estableciendo etiquetas semánticas como <div>, <h1>, y contenedores id para organizar y manipular los elementos mediante estilos y scripts. Esta base es esencial para una correcta manipulación del DOM.
- JavaScript fue fundamental para dotar de interactividad al calendario. Se utilizó para crear dinámicamente los días del mes, mediante métodos como createElement, appendChild y getElementById, permitiendo construir la interfaz de forma flexible y dinámica.
- Las hojas de estilo permitieron darle un diseño atractivo y ordenado al calendario. Mediante selectores como .class, #id y pseudoclases como :hover, se mejoró la experiencia visual y de uso, facilitando la comprensión y usabilidad del calendario por parte del usuario.



Evaluación del Aprendizaje

¿Cuál es la diferencia entre JavaScript y Jquery?

- JavaScript es un lenguaje de programación completo usado para crear funcionalidades en sitios web.
- jQuery es una biblioteca escrita en JavaScript que simplifica tareas comunes como manipular el DOM, manejar eventos y hacer peticiones AJAX.

¿Cuál es la estructura básica de una función con JavaScript? Coloque un ejemplo

```
function saludar(nombre) {  
    console.log("Hola, " + nombre);  
}  
saludar("Alexis");
```

¿Qué propiedad de CSS se emplea para definir el espacio entre caracteres en el texto?

La propiedad es *letter-spacing*.

Criterio	Excelente (5)	Bueno (4)	Satisfactorio (3)	Necesita Mejora (2)	Insuficiente (1)
Estructura del Código	Código bien organizado, limpio y fácil de entender	Código organizado con algunos errores menores	Código funcional, pero con algunos problemas de organización	Código desordenado con problemas evidentes	Código desorganizado o difícil de entender
Funcionalidad	Todas las funcionalidades requeridas están implementadas correctamente	La mayoría de las funcionalidades están implementadas correctamente	Algunas funcionalidades están implementadas, con errores menores	Varias funcionalidades faltan o no funcionan como se espera	Muchas funcionalidades faltan o no funcionan
Calidad del Código	Código bien escrito, eficiente y sin errores	Código eficiente con algunos errores menores	Código con algunos errores que afectan la eficiencia	Código con varios errores que afectan el rendimiento	Código ineficiente o lleno de errores
Uso de Tecnologías y Herramientas	Uso avanzado de tecnologías y herramientas adecuadas	Uso adecuado de tecnologías y herramientas con algunas limitaciones	Uso básico de tecnologías y herramientas, con errores menores	Uso limitado de tecnologías y herramientas, con errores significativos	Uso inapropiado o inexistente de tecnologías y herramientas
Experiencia del Usuario (UX)	Experiencia de usuario excelente, intuitiva y bien diseñada	Buena experiencia de usuario con algunos problemas menores	Experiencia de usuario aceptable, con problemas evidentes	Experiencia de usuario deficiente con problemas significativos	Experiencia de usuario mala, no intuitiva o muy confusa

Bibliografía

Heurtel Oliver (2023). *Php y mysql. Domine el desarrollo de un sitio web dinámico e interactivo*. España: ENI



Tecnológico Universitario **Vida Nueva**



Aubry Christopher (2021) *Html5 y css3. Domine los estándares de la creación de sitios*

web España ENI

Rúbricas/Listas de cotejo



TECNOLÓGICO UNIVERSITARIO VIDA NUEVA

RÚBRICA PARA TRABAJOS - CONSULTAS

CONSIDERACIONES	2p MUY BUENO	1p BUENO	0p REGULAR
Formato	La presentación tiene un formato uniforme. Permite la integración con cualquier otra parte sin necesidad de cambiar el aspecto.	Hay algunos párrafos o apartados que no se ajustan al formato. Tamaños de letras, viñetas, disposición del texto y gráficos.	El formato no es uniforme, contiene diferentes colores, diferente tamaño de letra, diferente tipo de letra, no contiene gráficos.
Relación con la temática	Refleja la intención y el contenido se relaciona con la temática	Refleja la intención y, pero el contenido no se relaciona con la temática	No refleja la intención y el contenido no se relaciona con la temática
Nivel de argumentación	Sustenta las ideas con citas tomadas de documentos académicos con alto rigor académico	Sustenta las ideas con citas tomadas de documentos académicos	Las ideas presentadas carecen de sustento académico, no se respaldan en citas.
Redacción	La redacción es clara, gramaticalmente correcto y tiene buena forma de presentación, se hace uso de la terminología de la asignatura.	La redacción es clara, pero no es gramaticalmente correcto, o no tiene buena forma de presentación. Hay un uso aceptable de conceptos de la asignatura.	Es difícil de entender lo redactado, es pobre la calidad gramatical, o no se hace uso de los conceptos propios de la asignatura.
Usos de gráficos	Los gráficos y figuras están correctamente utilizados, tienen sentido, ayudan a entender la explicación.	Falta algún grafico o figura que ayude a entender mejor un concepto. Algunas de las figuras son innecesarias.	No se utilizan gráficos ni figuras o las que se utilizan no aclaran nada.



TECNOLÓGICO UNIVERSITARIO VIDA NUEVA

RÚBRICA PARA FOROS

CONSIDERACIONES	2p MUY BUENO	1p BUENO	0p REGULAR
Relación con la temática	Refleja la intención y el contenido se relaciona con la temática	Refleja la intención y, pero el contenido no se relaciona con la temática	No refleja la intención y el contenido no se relaciona con la temática
Nivel de argumentación	Sustenta las ideas con citas tomadas de documentos académicos con alto rigor académico	Sustenta las ideas con citas tomadas de documentos académicos	Las ideas presentadas carecen de sustento académico, no se respaldan en citas.
Interacción	Respeta las opiniones de los demás. Sabe retomar intervenciones y aporta nuevos elementos.	Responde de manera sistemática a las opiniones de los demás, no aporta elementos nuevos.	No se incorpora al diálogo y expresa opiniones independientes.
Redacción	La redacción es clara, gramaticalmente correcto y tiene buena forma de presentación, se hace uso de la terminología de la asignatura.	La redacción es clara, pero no es gramaticalmente correcto, o no tiene buena forma de presentación. Hay un uso aceptable de conceptos de la asignatura.	Es difícil de entender lo redactado, es pobre la calidad gramatical, o no se hace uso de los conceptos propios de la asignatura.
Discusión	Estimula y promueve a los demás para profundizar en la discusión.	El autor plantea ideas de discusión que no profundizan lo que se debate.	El autor se limita a decir lo que opina, no genera discusión.



TECNOLÓGICO UNIVERSITARIO VIDA NUEVA

RÚBRICA DEL TRABAJO FINAL DE ASIGNATURA

INFORME/0.5:

CONSIDERACIONES	0,10 MUY BUENO	0, 05 BUENO	0, 03 REGULAR
Ortografía	No existen errores ortográficos en el informe.	Se detecta un error ortográfico en el informe	Se detecta dos o tres errores ortográficos en el informe
Contenido	Toda la información en el informe es coherente con el tema planteado	La información es parcialmente coherente con el tema planteado	La información no es coherente con el tema planteado.
Actividades	Presenta evidencia de las actividades realizadas en la ejecución.	Las evidencias presentadas de las actividades son escasas.	Las evidencias presentadas no tienen relación con las actividades.
Conclusiones	Las conclusiones son claras y están en relación a los objetivos	Las conclusiones son claras, pero no están en relación a los objetivos	Las conclusiones no son claras y no están en relación a los objetivos
Referencias bibliográficas	Las referencias bibliográficas son adecuadas y tienen relación con el tema planteado.	Existen varias referencias bibliográficas que no tienen relación con el tema planteado.	No presenta referencias bibliográficas o ninguna tiene relación con el tema planteado.



TECNOLÓGICO UNIVERSITARIO VIDA NUEVA

RÚBRICA PARA LECCIONES ORALES - PARTICIPACIONES

CONSIDERACIONES	2p MUY BUENO	1p BUENO	0p REGULAR
Dominio del tema	Demuestra un dominio profundo del contenido	Demuestra dominio del contenido	Demuestra desconocimiento del contenido
Nivel de argumentación	Sustenta las ideas con referencias a los documentos trabajados en clase y refuerza con referencias complementarias	Sustenta las ideas con referencias a los documentos trabajados en clase, pero no refuerza con referencias complementarias	Las ideas presentadas carecen de sustento académico.
Seguridad	Demuestra seguridad durante toda la intervención	Demuestra seguridad parcialmente en la intervención	Se muestra muy inseguro durante la intervención
Claridad – dicción	Pronuncia las palabras correctamente, es sencillo entender el tema que se trata.	Pronuncia algunas palabras de forma incorrecta, pero en general se entiende el tema que se trata.	Tiene dificultad para articular las palabras, por lo que resulta difícil entender el tema que se trata.
Respuestas	Responde todas las preguntas de manera correcta	Responde algunas preguntas de manera correcta	Las respuestas a las preguntas son incorrectas.

Tareas y Evaluaciones



TRABAJO: AUTÓNOMO COLABORATIVO

TEMA: Informe formulario web del IMC VALIDACION

Nombre: Stalin Carrión	Nivel: Cuarto
Carrera: Desarrollo de Software	Jornada: Matutina
Docente: Ing. Jorge Beltrán	Código: 25251781
Fecha: 19/4/2025	Calificación: _____

Informe del funcionamiento de la Calculadora de IMC

La aplicación web desarrollada tiene como propósito principal calcular el Índice de Masa Corporal (IMC) de una persona en base a los datos ingresados en un formulario. El IMC es una medida útil para evaluar si una persona tiene un peso adecuado en relación con su altura. La estructura general de esta calculadora está conformada por un archivo HTML que contiene el formulario y una tabla para mostrar los resultados, además de un archivo JavaScript que maneja la lógica de cálculo y presentación dinámica.

En el código HTML, se presenta un formulario que solicita al usuario cinco datos fundamentales: nombre, apellido, edad, talla (en metros) y peso (en kilogramos). Todos los campos son obligatorios, y algunos cuentan con restricciones de valor mínimo y máximo para asegurar que la información ingresada sea razonable. Por ejemplo, la altura debe estar entre 0.5 m y 2.5 m, mientras que el peso debe oscilar entre 1 kg y 300 kg. También se incluye un botón estilizado con una clase 'btn-grad', que activa el cálculo del IMC una vez que se presiona.

En cuanto al código JavaScript, la lógica principal se activa cuando el documento ha cargado completamente. Se asigna un *event listener* al formulario, el cual intercepta el evento 'submit'.

Este evento es cancelado mediante `preventDefault()` para evitar que la página se recargue, y en su lugar, se captura la información ingresada. Luego de validar que los campos no estén vacíos y que los datos numéricos sean correctos, se procede a realizar el cálculo del IMC utilizando la fórmula tradicional: peso dividido por la altura al cuadrado.

Una vez calculado el IMC, se clasifica dentro de una de las siguientes categorías: “Bajo peso” (IMC menor a 18.5), “Normal” (entre 18.5 y 24.9), “Sobrepeso” (entre 25 y 29.9) u “Obesidad” (30 o más). Esta información se organiza en una fila de tabla que se añade dinámicamente al cuerpo de la tabla (`<tbody>`), la cual inicialmente está oculta mediante una clase CSS (`hidden`). Si la tabla estaba invisible, se le remueve la clase para hacerla visible. Finalmente, el formulario se limpia automáticamente para permitir nuevos cálculos.

Esta calculadora de IMC es una herramienta sencilla, intuitiva y funcional que permite al usuario obtener información inmediata sobre su estado nutricional con tan solo ingresar algunos datos personales. La implementación está bien estructurada, combinando HTML para la interfaz, JavaScript para la lógica y un archivo CSS externo para la apariencia visual.

Anexos

Fecha	Título	Código	Asignatura	Tipo
2025-04-19 10:22:04	JAVASCRIPT. APRENDEA PROGRAMAR EN EL LENGUAJE DE LA WEB D5173	25251781	PROGRAMACIÓN WEB	LIBRO DIGITAL

Calculadora de IMC

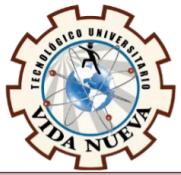
Nombre
Apellido
Edad
Talla (m)
Peso (kg)

CALCULAR IMC

Nombre	Apellido	Edad	Talla (m)	Peso (kg)	IMC	Categoría
Wendy	Carrión	19	1.6	52	20.31	Normal
rodolfo	suarez	50	1.79	90	28.09	Sobrepeso



```
1 document.addEventListener("DOMContentLoaded", () => {
2   const form = document.getElementById("imc-form");
3   const tablaIMC = document.getElementById("tabla-imc");
4   const resultadoIMC = document.getElementById("resultado-imc");
5
6   form.addEventListener("submit", (event) => {
7     event.preventDefault();
8
9     // Obtener los valores del formulario
10    const nombre = document.getElementById("nombre").value.trim();
11    const apellido = document.getElementById("apellido").value.trim();
12    const edad = parseInt(document.getElementById("edad").value);
13    const talla = parseFloat(document.getElementById("talla").value);
14    const peso = parseFloat(document.getElementById("peso").value);
15
16    // Validar los datos
17    if (!nombre || !apellido || isNaN(edad) || isNaN(talla) || isNaN(peso)) {
18      alert("Por favor, complete todos los campos correctamente.");
19      return;
20    }
21
22    // Calcular el IMC
23    const imc = (peso / (talla * talla)).toFixed(2);
24
25    // Determinar la categoría del IMC
26    let categoria = "";
27    if (imc < 18.5) {
28      categoria = "Bajo peso";
29    } else if (imc >= 18.5 && imc < 24.9) {
30      categoria = "Normal";
31    } else if (imc >= 25 && imc < 29.9) {
32      categoria = "Sobrepeso";
33    } else {
34      categoria = "Obesidad";
35    }
36
37    // Mostrar los resultados en la tabla
38    const nuevaFila = document.createElement("tr");
39    nuevaFila.innerHTML = `
40      <td>${nombre}</td>
41      <td>${apellido}</td>
42      <td>${edad}</td>
43      <td>${talla}</td>
44      <td>${peso}</td>
45      <td>${imc}</td>
46      <td>${categoria}</td>
47    `;
48    tablaIMC.appendChild(nuevaFila);
49
50    // Mostrar la tabla si está oculta
51    resultadoIMC.classList.remove("hidden");
52
53    // Limpiar el formulario
54    form.reset();
55  });
56 });
57
```



TRABAJO: AUTÓNOMO COLABORATIVO

TEMA: Desarrollar un formulario web del IMC para el cálculo de asistencia asignando un estado aprobado y reprobado de acuerdo al 80% como mínimo de aprobación.

Nombre: Stalin Carrión	Nivel: Tercero
Carrera: Desarrollo de Software	Jornada: Matutina
Docente: Ing. Jorge Beltrán	Código: 25251781
Fecha: 24/4/2025	Calificación: _____

El script comienza obteniendo una referencia al formulario HTML mediante su ID estudiante-form, usando document.getElementById(). Esto permite asociarle un evento cuando el usuario lo envíe. Luego, se selecciona el tbody de la tabla con querySelector("#tabla-estudiantes tbody"), ya que es ahí donde se insertarán las nuevas filas con los datos. También se crea un arreglo vacío llamado datos que almacenará los objetos de tipo estudiante con la información ingresada por el usuario.

Se agrega un "listener" al formulario que se activa cuando este se envía (submit). El evento se previene (event.preventDefault()) para evitar que la página se recargue al enviar. Luego, se recogen los valores de los inputs del formulario: nombre, apellido, asignatura, créditos y faltas. Los campos de texto se transforman a mayúsculas usando toUpperCase() y se eliminan espacios extras con trim(). Los valores numéricos son convertidos a enteros usando parseInt().

Antes de continuar, el código realiza validaciones sobre los valores numéricos. Verifica que el número de créditos esté entre 1 y 10, y que el número de faltas esté entre 0 y 10. Si los valores no

están dentro de los rangos permitidos, muestra un mensaje de alerta con alert() y detiene la ejecución de la función con return.

Cuando los datos son válidos, se crea un objeto estudiante con las propiedades nombre, apellido, asignatura, creditos y faltas. Este objeto se agrega al arreglo datos con push(). Después de eso, se llama a la función agregarFila() para mostrar los datos del nuevo estudiante en la tabla. Finalmente, se limpian los inputs del formulario con limpiarInputs() para dejar el formulario listo para otro ingreso.

El script contiene dos funciones para calcular los porcentajes de asistencia e inasistencia. La función calcularPorcentajeInasistencia() toma en cuenta que cada crédito representa 20 horas, y cada falta equivale a 2 horas perdidas. Calcula la cantidad total de horas y las horas perdidas, luego obtiene el porcentaje correspondiente. La función calcularPorcentajeAsistencia() simplemente resta ese porcentaje a 100. Ambas devuelven los valores redondeados a dos decimales.

La función agregarFila() se encarga de crear una nueva fila HTML (<tr>) con los datos del estudiante y los porcentajes calculados. También incluye un botón "Eliminar". La fila se inserta al final del tbody de la tabla con appendChild(). Además, se asigna un evento a ese botón para que, al hacer clic, se elimine la fila con la función eliminarFila().

La función limpiarInputs() simplemente limpia el valor de cada campo de entrada del formulario, dejándolos vacíos después de agregar un estudiante.

La función eliminarFila() recibe como parámetros la fila HTML y el objeto estudiante. Primero remueve visualmente la fila de la tabla con remove(). Luego, busca la posición del objeto dentro del array datos usando indexOf() y lo elimina del arreglo con splice(), asegurando que tanto la vista como los datos internos estén sincronizados.

Anexos

The screenshot shows a digital book interface with the following details:

Header: 2025-04-24, EL GRAN LIBRO DE HTML5, CSS3 Y JAVASCRIPT D3089, Autor: GAUCHAT, J.D., Digital, 25251781, PROGRAMACIÓN WEB, LIBRO DIGITAL.

Form (Cálculo de Asistencia):

Nombre	Apellido	Asignatura	Créditos
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Inputs:

- ① Faltas:
- Enviar:

Results (Resultados):

Nombre	Apellido	Asignatura	% Porcentaje de Asistencia	① Porcentaje de Insistencia	
STALIN	CARRION	PROGRAMACION	92.50%	7.50%	<input type="button" value="Eliminar"/>
LUCHITO	PEREZ	BASE DE DATOS	94.29%	5.71%	<input type="button" value="Eliminar"/>
DAYANA	GONZALES	NETWORKING	97.14%	2.86%	<input type="button" value="Eliminar"/>

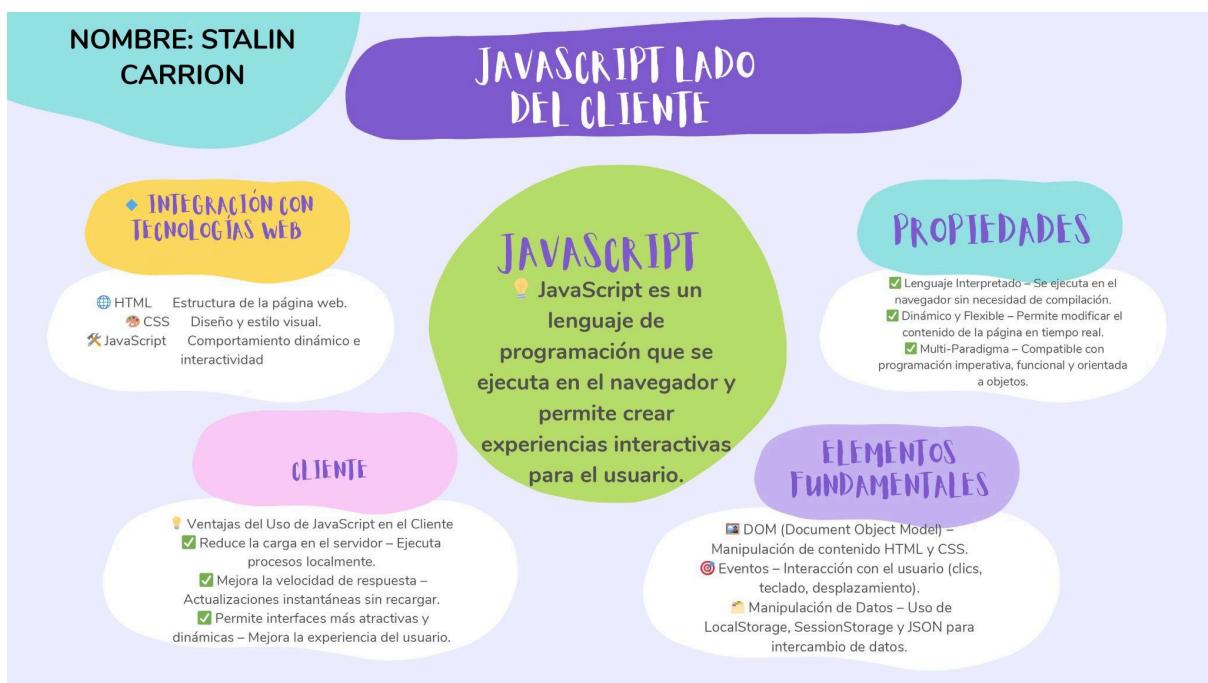
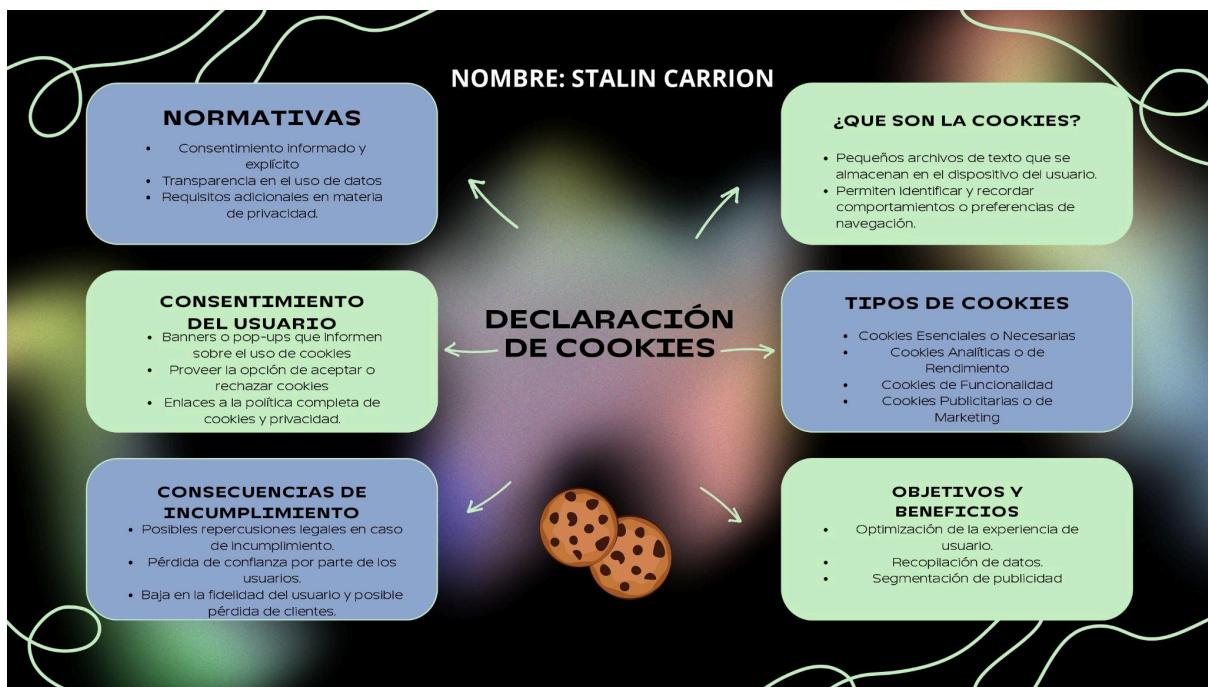


```
1 //funcion para calcular el porcentaje de inasistencia
2 function calcularPorcentajeInasistencia(faltas, creditos) {
3     const horasPorCredito = 20; // horas por crédito
4     const horasPorFalta = 2; // horas por falta
5     const totalHoras = horasPorCredito * creditos;
6     const horasFaltadas = horasPorFalta * faltas;
7     const porcentajeInasistencia = (horasFaltadas / totalHoras) * 100;
8     return porcentajeInasistencia.toFixed(2); // Redondear a 2 decimales
9 }
10
11
12 //funcion para calcular porcentaje de asistencia
13 function calcularPorcentajeAsistencia(faltas, creditos) {
14     const porcentajeInasistencia = calcularPorcentajeInasistencia(faltas, creditos);
15     const porcentajeAsistencia = 100 - porcentajeInasistencia;
16     return porcentajeAsistencia.toFixed(2); // Redondear a 2 decimales
17 }
18
19 //funcion para agregar una fila a la tabla
20 function agregarFila(estudiante) {
21     const fila = document.createElement("tr");
22     fila.innerHTML =
23         ` ${estudiante.nombre}</td> 24         <td>${estudiante.apellido}</td> 25         <td>${estudiante.asignatura}</td> 26         <td>${calcularPorcentajeAsistencia(estudiante.faltas, estudiante.creditos)}%</td> 27         <td>${calcularPorcentajeInasistencia(estudiante.faltas, estudiante.creditos)}%</td> 28         <td><button class="btn btn-danger btn-sm eliminar">Eliminar</button></td>`; 29     tabla.appendChild(fila); 30 31     const botonEliminar = fila.querySelector(".eliminar"); 32     botonEliminar.addEventListener("click", function () { 33         eliminarFila(fila, estudiante); 34     }); 35 } 36 37 function limpiarInputs() { 38     //limpio los inputs 39     document.getElementById("nombre").value = ""; 40     document.getElementById("apellido").value = ""; 41     document.getElementById("asignatura").value = ""; 42     document.getElementById("creditos").value = ""; 43     document.getElementById("faltas").value = ""; 44 } 45 46 function eliminarFila(fila, estudiante) { 47     // Eliminar la fila de la tabla 48     fila.remove(); 49     // Eliminar el objeto del array datos 50     const index = datos.indexOf(estudiante); 51     if (index > -1) { 52         datos.splice(index, 1); 53     } 54 } 55 |
```

```
● ● ●

1 const formulario = document.getElementById("estudiante-form");
2 /* en una variable guardo el formulario para
3 que se active el evento con submit */
4 //tabla tendra la responsabilidad de mostrar los datos
5 // en la tabla, por lo que se le asigna el id de la tabla
6 const tabla = document.querySelector("#tabla-estudiantes tbody");
7 /* tbody es el cuerpo de la tabla, donde se van a ir
8 agregando los datos, por lo que se le asigna el id de la tabla*/
9 //creo un array vacio para guardar los datos
10 const datos = []; //array vacio para guardar los datos
11 //agrego un evento al formulario para que se active al enviar
12
13 formulario.addEventListener("submit", function (event) {
14     event.preventDefault(); //previene el comportamiento por defecto del formulario
15     //obtengo los valores de los inputs
16     const nombre = document.getElementById("nombre").value.trim().toUpperCase();
17     const apellido = document
18         .getElementById("apellido")
19         .value.trim()
20         .toUpperCase();
21     const asignatura = document
22         .getElementById("asignatura")
23         .value.trim()
24         .toUpperCase();
25     const numeroCreditos = parseInt(document.getElementById("creditos").value);
26     const numeroFaltas = parseInt(document.getElementById("faltas").value);
27
28     if (isNaN(numeroCreditos) || numeroCreditos < 1 || numeroCreditos > 10) {
29         alert("El número de créditos debe estar entre 1 y 10.");
30         return;
31     }
32
33     if (isNaN(numeroFaltas) || numeroFaltas < 0 || numeroFaltas > 10) {
34         alert("El número de faltas debe estar entre 0 y 10.");
35         return;
36     }
37
38     //creo un objeto con los datos del formulario
39     const estudiante = {
40         nombre: nombre,
41         apellido: apellido,
42         asignatura: asignatura,
43         creditos: numeroCreditos,
44         faltas: numeroFaltas,
45     };
46     //agrego el objeto al array inventario
47     datos.push(estudiante);
48     //llamo a la funcion para mostrar los datos en la tabla
49     agregarFila(estudiante);
50     //limpio los inputs
51     limpiarInputs();
52
53 });


```





TRABAJO: AUTÓNOMO COLABORATIVO

TEMA: Informe sobre el proceso de insertar en mvc

Nombre: Stalin Carrión	Nivel: Tercero
Carrera: Desarrollo de Software	Jornada: Matutina
Docente: Ing. Jorge Beltrán	Código: 25251781
Fecha: 13/05/2025	Calificación: _____

Análisis

formulario.php

Este archivo es una vista que genera un formulario HTML para que el usuario pueda ingresar datos relacionados con un "Departamento". Su propósito principal es capturar información como el nombre, ubicación y área del departamento, y enviarla al controlador correspondiente para su procesamiento.

1. Estructura del formulario: El formulario utiliza el método 'POST' para enviar los datos al archivo `departamentoController.php`, ubicado en el directorio controllers. Incluye campos de entrada para el nombre, ubicación y área, además de un campo oculto ('hidden') que define la acción como "crear".
2. Interacción: Cuando el usuario presiona el botón "Guardar", los datos se envían al controlador especificado en el atributo 'action'.



```
1 <body>
2   <div class="form-container">
3     <h2 class="text-center mb-4">Formulario de Departamento</h2>
4     <form action="../controllers/departamentoController.php" method="POST">
5
6       <div class="mb-3">
7         <label for="nombre" class="form-label">Nombre</label>
8         <input type="text" class="form-control" id="nombre" name="nombre" maxLength="30">
9       </div>
10      <div class="mb-3">
11        <label for="ubicacion" class="form-label">Ubicación</label>
12        <input type="text" class="form-control" id="ubicacion" name="ubicacion" maxLength="30">
13      </div>
14      <div class="mb-3">
15        <label for="area" class="form-label">Área</label>
16        <input type="text" class="form-control" id="area" name="area" maxLength="30">
17      </div>
18
19      <!--hidden-->
20      <input type="hidden" name="accion" value="crear">
21
22      <!--boton de enviar el formulario-->
23      <button type="submit" class="btn btn-primary w-100">Guardar</button>
24    </form>
25  </div>
26 </body>
```

Archivo `Departamento.php`

Se incluye en `formulario.php` mediante `require_once`. Esto sugiere que `Departamento.php` es un modelo que define la estructura y lógica relacionada con los departamentos.

Este archivo contiene una clase `Departamento` con propiedades como `nombre`, `ubicacion` y `area`, además de métodos para interactuar con la base de datos (por ejemplo, crear, leer, actualizar y eliminar departamentos).

Se utiliza directamente en el formulario, su inclusión es necesaria para validar o procesar datos relacionados con los departamentos.



```

1 <?php
2
3 require_once __DIR__ . '/../config/database.php';
4
5
6 class Departamento
7 {
8     //variable co la isntancia dcla conejon
9     private $db;
10
11    //funciñm por medio de db que pemrita qçacerder a laq combexiuon esto con una imagne
12    public function __construct()
13    {
14        $this->db = database::conectar();
15    }
16
17    public function crear($data)
18    {
19        // Prepara la consulta SQL para insertar un nuevo departamento
20        $stmt = $this->db->prepare("INSERT INTO departamento ( nombre, ubicacion, area) VALUES (?, ?, ?)");
21        return $stmt->execute([$data['nombre'], $data['ubicacion'], $data['area']]);
22    }
23}
24

```

Archivo `departamentoController.php`

Este archivo, ubicado en el directorio controllers, es el controlador al que se envían los datos del formulario. Su función principal es recibir los datos enviados por el formulario, procesarlos y realizar las acciones necesarias, como guardar un nuevo departamento en la base de datos.

Este archivo lee el valor del campo oculto `accion` para determinar qué operación realizar (en este caso, "crear").

- Interacción con el modelo: Utiliza el modelo 'Departamento.php' para realizar operaciones en la base de datos, como insertar un nuevo registro.
- Redirección o respuesta: Después de procesar los datos, este archivo podría redirigir al usuario a otra página (por ejemplo, una lista de departamentos) o mostrar un mensaje de éxito o error.



```
1 <?php
2 require_once __DIR__ . '/../models/Departamento.php';
3
4 $departamento = new Departamento();
5
6 if (isset($_POST['accion']) && $_POST['accion'] == 'crear') {
7     $departamento->crear($_POST);
8 }
9
10 ?>
```

Archivo database.php

Este archivo se encuentra en el directorio config y tiene como objetivo principal gestionar la conexión a la base de datos del sistema. Es una clase estática que proporciona un método para establecer una conexión con la base de datos utilizando PDO (PHP Data Objects).

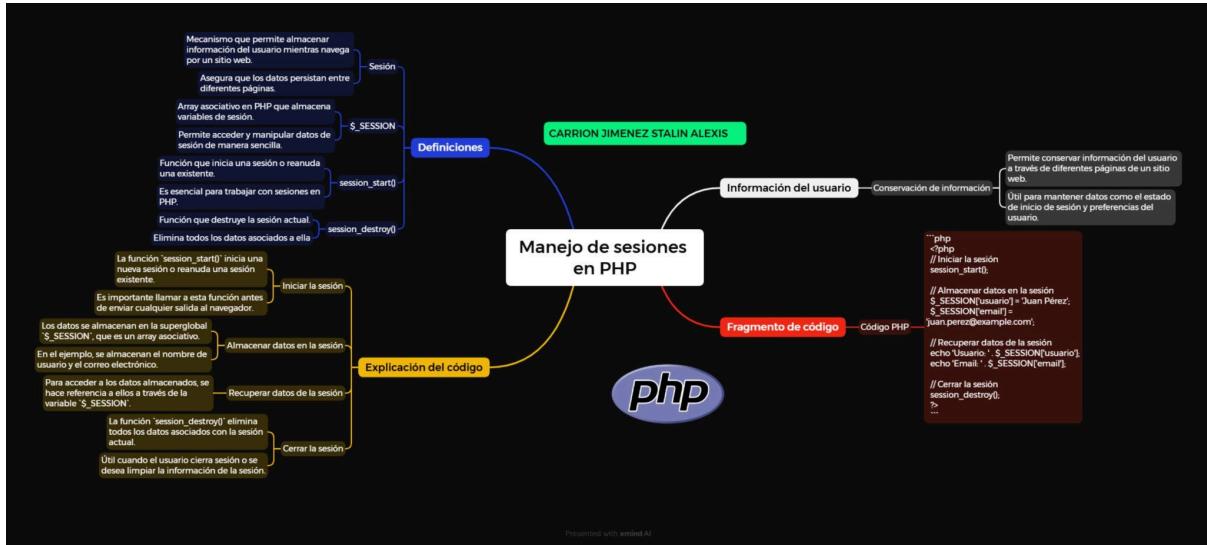


```
1 <?php
2
3 class database
4 {
5     public static function conectar()
6     {
7         // Conexión a la base de datos
8         // Cambia los parámetros según tu configuración
9         // Cambia 'localhost', 'SistemaRol2', 'root' y '' según tu configuración de base de datos
10        return new PDO('mysql:host=localhost;dbname=sistemarol', 'root', '');
11    }
12 }
13
```

Flujo general del sistema

1. El usuario accede a 'formulario.php', donde se muestra el formulario para ingresar datos de un departamento.
2. Al enviar el formulario, los datos se envían a 'departamentoController.php' mediante el método 'POST'.
3. El controlador procesa los datos, interactúa con el modelo 'Departamento.php' para realizar la operación solicitada (por ejemplo, guardar un nuevo departamento en la base de datos) y devuelve una respuesta al usuario.
4. Una vez hecha la solicitud procesa la consulta sql que inserta un nuevo dato en la base de datos, este proceso se puede repetir muchas veces ya que el id es autoincrementable por lo cual no hay problema.

Anexos





TRABAJO: AUTÓNOMO COLABORATIVO

TEMA: Informe sobre el proceso de mostrar en mvc

Nombre: Stalin Carrión	Nivel: Cuarto
Carrera: Desarrollo de Software	Jornada: Matutina
Docente: Ing. Jorge Beltrán	Código: 25251781
Fecha: 15/05/2025	Calificación: _____

Informe

Este código tiene como objetivo principal mostrar una lista de departamentos que vienen desde una base de datos. Lo primero que se hace es incluir el archivo del modelo Departamento.php, que contiene la lógica para conectarse y trabajar con la base de datos. Luego, se crea una instancia (u objeto) de esa clase y se llama al método obtenerTodos(), que hace una consulta SELECT * para traer todos los departamentos guardados. Los datos que se obtienen se guardan en la variable \$departamento.

```

1 <?php
2 require_once '../models/Departamento.php';
3 $departamento = new Departamento();
4 $departamento = $departamento->obtenerTodos
5 ();
?>

```

Después de obtener los datos, el código empieza a construir la interfaz con HTML y Bootstrap.

Se crea una tarjeta (card) dentro de un contenedor, donde se coloca el título "Listado de Departamentos" y un botón que lleva a un formulario para crear un nuevo departamento. Todo esto tiene un diseño oscuro y moderno gracias al estilo CSS personalizado que se encuentra dentro del <style>.



The screenshot shows a code editor with a dark theme. At the top, there are three colored circular icons: red, yellow, and green. Below them is the CSS code for a card component:

```
1 .card {  
2     background-color: #2c2c2c;  
3     border-radius: 12px;  
4     box-shadow: 0 0 15px rgba(0, 0, 0, 0.5);  
5 }  
6  
7 .card-header {  
8     background-color: #4CAF50;  
9     color: white;  
10    font-weight: bold;  
11    border-radius: 12px 12px 0 0;  
12    padding: 15px;  
13 }  
14  
15
```

Dentro del cuerpo de la tarjeta hay una tabla. Esta es la parte más importante porque es donde se van a mostrar los datos. Aquí se usa PHP para verificar si hay datos en \$departamento. Si sí hay datos, se usa un foreach para recorrer cada elemento (cada departamento), y se imprime una fila (<tr>) con su respectivo ID, nombre, ubicación y área. Además, se incluyen dos botones: uno para editar (que manda al formulario con el ID por la URL) y otro para eliminar (aunque ese todavía no hace nada porque no está enlazado a ningún archivo real).



```

1 <tbody>
2     <?php if (!empty($departamento)) : ?>
3         <?php foreach ($departamento as $dep) : ?>
4             <tr>
5                 <td><?= htmlspecialchars($dep['id_departamento']) ?>
6                     <td><?= htmlspecialchars($dep['nombre']) ?></td>
7                     <td><?= htmlspecialchars($dep['ubicacion']) ?></td>
8                     <td><?= htmlspecialchars($dep['area']) ?></td>
9                     <td class="text-center">
10                         <a href="formulario.php?id=<?= $dep['id_departamen
11                             to'] ?>" class="btn btn-warning btn-sm">Editar</a>
12                         <a href="#" class="btn btn-danger btn-sm">Eliminar
13                     </a>
14                 </td>

```

En caso de que no haya ningún departamento en la base de datos, el sistema muestra un solo mensaje dentro de la tabla que dice: "No hay departamentos registrados." Esto es importante porque evita que la tabla quede vacía sin que el usuario sepa por qué.



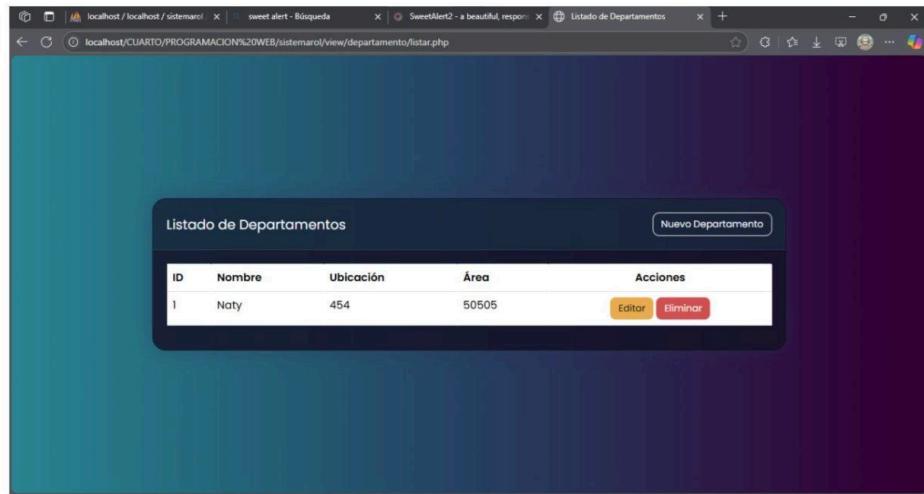
```

1 <?php else : ?>
2     <tr>
3         <td colspan="5" class="text-center text-warning">
4             No hay departamentos registrados.</td>
5         </tr>
6     <?php endif; ?>

```

Este código hace lo necesario para traer datos de la base de datos y mostrarlos de forma ordenada en una tabla. Usa PHP para la parte lógica y HTML con Bootstrap para el diseño visual. El

proceso de mostrar se basa en verificar si hay datos, recorrerlos y pintarlos uno por uno en pantalla. Es un ejemplo claro de cómo separar el backend (datos y lógica) del frontend (interfaz visual).



Anexos

