



1. Datos Generales de la asignatura

| | |
|---------------------------------|-------------------------------|
| Nombre de la asignatura: | Desarrollo de Plataformas Web |
| Clave de la asignatura: | TAD-2502 |
| SATCA¹: | 2-3-5 |
| Carrera: | Ingeniería Informática |

2. Presentación

Caracterización de la asignatura

El Internet ha venido a revolucionar los sistemas de información empresariales. Hoy en día los lenguajes de programación orientados a la Web son cada vez más variados y especializados que implican estar al día sobre los mismos. El MVC es una arquitectura de software que estructura la lógica de forma eficiente y escalable. Algunos frameworks que lo consideran son Django, Ruby on Rails, Sails, Laravel, entre otros.

Esta materia aportará al perfil profesional del ingeniero informático las competencias siguientes:

Habilidad para obtener y procesar información relacionada con los requerimientos informáticos para el desarrollo de aplicaciones en la Web.

Capacidad para analizar y diseñar sistemas de información complejos que se ejecutan en Internet.

Capacidad de implementar sistemas de información de acuerdo con la tecnología actual.

Habilidad en el uso de herramientas CASE en el desarrollo de proyectos informáticos Web.

Capacidad para adaptarse a los cambios tecnológicos en relación con entornos de programación Web para el desarrollo de sistemas.

La asignatura permite adquirir conocimientos para desarrollar un sistema de información que dé solución a los requerimientos de los usuarios en un ambiente Web.

Esta asignatura se relaciona con la materia de Desarrollo de Aplicaciones Web y Programación Orientada a Objetos en todos sus temas.

¹ Sistema de Asignación y Transferencia de Créditos Académicos





Desarrollo e Implementación de sistemas de información en la aplicación de conocimientos de herramientas CASE para el desarrollo de sistemas de información y técnicas de uso de métricas.

En Tópicos de Bases de Datos para definir la arquitectura de un sistema y elaborar consultas y operaciones sobre el mismo.

En la asignatura Programación en Ambiente Cliente Servidor, en conocer, diseñar y desarrollar aplicaciones atendiendo la arquitectura cliente servidor.

También en la Calidad en los Sistemas de información en aplicar normas y estándares de calidad en el entorno del desarrollo de sistemas de información.

Intención didáctica

La forma de abordar los contenidos de la asignatura será a través de exposición de los temas con ejemplos, investigaciones, realización de prácticas y desarrollo paulatino de un proyecto integrador real que resuelva un sistema de información en la Web bajo la arquitectura MVC.

Los temas deberán ser tratados con un enfoque actualizado de acuerdo con las nuevas tecnologías.

En el primer tema se abordarán temas del análisis y diseño del problema a resolver, instalación y configuración del framework y la arquitectura de desarrollo basada en MVC. Además, se explorarán los conocimientos de interfaz Web de carga asíncrona y su implementación en el desarrollo de soluciones.

En el segundo tema se aborda **GitHub** para adquirir los conocimientos, habilidades y actitudes que los estudiantes o profesionales para dominar la herramienta, y aplicar el desarrollo de proyectos web y la colaboración en equipo.

En el tercer tema se exploran los conocimientos para implementar los roles y permisos de usuarios para mantener la seguridad del sistema a través del uso de sesiones.

En el cuarto tema se tratan los temas de cookies y ftp e informes que son fundamentales en





los sistemas en la Web. En este tema se generan también los informes en distintos formatos PDF, EXCEL y CSV así como la publicación del proyecto en un sitio Web.

El estudiante debe desarrollar habilidades en el trabajo colaborativo, crítico, síntesis y abstracción de información no estructurada, con la finalidad de que sea capaz de desarrollar sistemas de información aplicando la arquitectura MVC.

El profesor será un facilitador del conocimiento, proporcionando al estudiante ejemplos y prácticas que lo guíen en adquirir las competencias necesarias de la materia.

3. Participantes en el diseño y seguimiento curricular del programa

| Lugar y fecha de elaboración o revisión | Participantes | Observaciones |
|---|----------------------|---|
| Instituto Tecnológico de Cd. Guzmán. 3 de Abril 2025 | Raquel Ochoa Ornelas | Programa elaborado para la conformación de la especialidad “Tecnologías Emergentes en Software, Análisis de Datos y Aprendizaje Automático” |

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura

Desarrollar un sistema de información dinámico y especializado en entorno Web que resuelva un problema real de manera eficiente utilizando tecnologías de actualidad, tanto en el manejo de datos, diseño de interfaz y programación avanzada.

- Aplicar un lenguaje de programación basado en MVC para la Web.
- Utilizar herramientas CASE para el análisis y diseño de sistemas.
- Aplicar eficientemente las métricas en el desarrollo de sistemas Web.
- Diseñar variedad de informes completos y estructurados en el sistema Web.
- Adquirir experiencia en documentar sistemas de información Web.





5. Competencias previas

- Definir la arquitectura de un sistema de datos, elaborando consultas y operaciones sobre el mismo.
- Experiencia en ambiente cliente-servidor para diseñar y desarrollar sistemas de información en N-Capas atendiendo dicha arquitectura y conocimiento de protocolos de transferencia.
- Conocer y aplicar normas y estándares de calidad en el entorno del desarrollo de sistemas de información en la Web.

6. Temario

| No. | Temas | Subtemas |
|-----|--------------------------|---|
| 1 | Modelo Vista Controlador | 1.1 Definición, análisis y diseño del proyecto 1.2 Instalación y configuración del entorno de trabajo 1.3 Conceptos de programación orientada a objetos 1.4 Conexión e instrucciones SQL 1.5 Estructura de un proyecto en MVC 1.6 Configuración de rutas 1.7 Envío de datos a controladores 1.8 Envío de datos a modelos 1.9 Envío de datos a vistas 1.10 Implementación de objetos de acceso a datos 1.11 Implementación de Helpers 1.12 Implementación de Bootstrap 1.13 Introducción a las técnicas de carga asíncrona |
| 2 | Control de Versiones | 2.1 Introducción a GitHub 2.2 Uso básico de GitHub 2.3 Colaboración en GitHub 2.4 Avanzado CI/CD y Automatización con GitHub 2.5 Gestión de Proyectos con GitHub 2.6 Buenas prácticas y seguridad en GitHub |





| | | |
|---|--------------------------------------|--|
| 3 | Roles y Permisos de Usuarios | 3.1 Desarrollo de vista roles de usuarios 3.2 Guardar, consultar, actualizar, eliminar roles de usuarios 3.3 Crear vistas de usuarios 3.4 Guardar, consultar, actualizar, eliminar usuarios 3.5 Validar usuario y contraseña 3.6 Configurar y enviar correo para cambio de contraseña 3.7 Sesiones |
| 4 | Cookies, FTP, Informes y Publicación | 4.1 Establecer y leer una cookie 4.2 Caducidad y eliminación de una cookie 4.3 Descarga y carga de archivos con FTP 4.4 Eliminación de archivos con FTP 4.5 Creación y eliminación de directorios con FTP 4.6 Informes Excel, PDF y CSV 4.7 Publicación del sitio en un servidor Web o Hosting |

7. Actividades de aprendizaje de los temas

| 1. Modelo Vista Controlador | |
|---|--|
| Competencias | Actividades de aprendizaje |
| <p>Específica(s):</p> <p>Comprender el patrón MVC en un lenguaje específico para el desarrollo de la plataforma Web.</p> <p>Genéricas:</p> <p><u>Competencias instrumentales</u></p> <ul style="list-style-type: none"> Capacidad de lógica, pensamiento creativo. | 1.1 Definir el análisis y diseño del proyecto. 1.2 Instalar framework. 1.3 Investigar la arquitectura MVC. 1.4 Exponer fundamentos del MVC. 1.5 Desarrollar ejercicios y prácticas para aplicar los conceptos anteriormente vistos. 1.6 Implementar catálogos y movimientos en el proyecto en MVC. 1.7 Investigar los fundamentos de la carga asíncrona en aplicaciones Web. |





| 2. Control de Versiones | |
|--|---|
| Competencias | Actividades de aprendizaje |
| <p>Específica(s):</p> <p>Gestionar y colaborar eficientemente en proyectos de desarrollo de software utilizando Git y GitHub, aplicando prácticas de control de versiones, integración continua y despliegue automatizado, con el fin de optimizar el trabajo en equipo, garantizar la calidad del código y asegurar la seguridad en los repositorios.</p> <p>Genéricas:</p> <p><u>Competencias instrumentales</u></p> <ul style="list-style-type: none">• Capacidad de lógica y pensamiento creativo.• Capacidad de gestionar información <p><u>Competencias interpersonales</u></p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Habilidades interpersonales.• Trabajo en equipo.• Ética. | <p>2.1 Investigar los fundamentos de la carga asíncrona en aplicaciones Web.</p> <p>2.2 Los estudiantes instalarán Git en su máquina local y configurarán su cuenta de GitHub.</p> <p>2.3 Crear un repositorio en GitHub y clonarlo en su máquina local para comenzar a trabajar en un proyecto.</p> <p>2.4 Hacer un primer commit en el repositorio local y realizar un git push para cargar los cambios a GitHub.</p> <p>2.5 Los estudiantes deberán realizar cambios simultáneos en la misma línea de un archivo para crear un conflicto, y luego resolverlo usando los comandos apropiados.</p> <p>2.6 Realizar un fork de un repositorio de código abierto en GitHub, clonar el repositorio en su máquina local, y realizar una mejora o corrección, luego enviar un pull request.</p> <p>2.7 Revisar el código de un compañero usando un pull request, proporcionando comentarios constructivos sobre el código y sugiriendo mejoras.</p> <p>2.8 Crear un flujo de trabajo básico en GitHub Actions para ejecutar pruebas automáticas cada vez que se hace un commit en el repositorio.</p> <p>2.9 Configurar un flujo de trabajo en GitHub Actions para desplegar automáticamente la aplicación a un servicio como Heroku o Netlify después de cada commit.</p> |





| | |
|--|--|
| | <p>2.10 Crear un tablero de GitHub Projects para organizar las tareas del proyecto, asignar responsables y establecer fechas límite.</p> <p>2.11 Utilizar Dependabot para revisar y actualizar las dependencias de un proyecto, y asegurarse de que el código no tiene vulnerabilidades conocidas.</p> |
|--|--|

| 3. Roles y Permisos de Usuarios | |
|--|--|
| Competencias | Actividades de aprendizaje |
| <p>Específica(s):</p> <p>Implementa roles de usuarios y sesiones que permitan mantener la seguridad en la aplicación Web a desarrollar.</p> <p>Genéricas:</p> <p><u>Competencias instrumentales</u></p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de lógica y pensamiento creativo. • Capacidad de gestionar información <p><u>Competencias interpersonales</u></p> <ul style="list-style-type: none"> • Capacidad crítica y autocrítica. • Habilidades interpersonales. • Trabajo en equipo. • Ética. | <p>3.1 Desarrollar la vista de los roles y de usuarios.</p> <p>3.2 Investigar la importancia y funcionamiento de las variables de sesión en una aplicación Web.</p> <p>3.3 Implementar sesiones en un ejemplo base.</p> <p>3.4 Limitar una sesión de usuario activa.</p> <p>3.5 Recuperar contraseña.</p> <p>3.6 Realizar la validación de usuarios en el proyecto.</p> <p>3.7 Generar y validar Token para cambio de contraseña.</p> <p>3.8 Incluir el cambio de contraseña y confirmación desde el correo.</p> |





| 4. Cookies, FTP, informes y publicación | |
|---|--|
| Competencias | Actividades de aprendizaje |
| <p>Específica(s):</p> <p>Implementa el manejo de cookies, ftp y generación de informes en un sistema de información Web publicando en un servidor el proyecto final.</p> <p>Genéricas:</p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis.• Capacidad de lógica y pensamiento creativo.• Capacidad de gestionar información <p><u>Competencias interpersonales</u></p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Habilidades interpersonales.• Trabajo en equipo.• Ética.• Comunicación con otros profesionistas de otras áreas para la resolución de problemas. <p><u>Competencias sistemáticas</u></p> <ul style="list-style-type: none">• Habilidades de investigación.• Capacidad de aprender.• Habilidad de liderazgo. | <p>4.1 Investigar usos e implementación de FTP en aplicaciones Web para la carga y descarga de archivos.</p> <p>4.2 Implementar el uso de cookies.</p> <p>4.3 Realizar ejemplos del uso de ftp.</p> <p>4.4 Generar los informes del proyecto en distintos formatos.</p> <p>4.5 Investigar servidores gratuitos para crear el sitio Web.</p> <p>4.6 Seleccionar y justificar el servidor Web a utilizar.</p> <p>4.7 Publicar la aplicación del proyecto desarrollado durante el curso</p> |





8. Práctica(s)

1. Instalación del ambiente de programación.
2. Práctica de arquitectura MVC en un proyecto Web.
3. Práctica de instrucciones de E/S.
4. Práctica de estructuras de control.
5. Práctica de funciones del lenguaje.
6. Práctica con interacción de formularios.
7. Práctica de conexión a datos en un SGBD para la Web.
8. Prácticas de aplicaciones de acceso a datos utilizando la arquitectura MVC.
9. Prácticas de la implementación de Ajax en un proyecto.
10. Práctica de gestión de repositorios.
11. Práctica de colaboración en proyectos.
12. Práctica de automatización con GitHub Actions.
13. Práctica de gestión de proyectos con GitHub projects e Issues.
14. Práctica de seguridad y buenas prácticas en GitHub.
15. Práctica de forking y contribución a proyectos de código abierto.
16. Práctica del manejo de variables de sesión en el desarrollo de aplicaciones Web.
17. Práctica del uso de las cookies en una aplicación Web.
18. Práctica de la implementación de FTP en aplicaciones Web.
19. Prácticas de generación de informes en distintos formatos.
20. Práctica de pasos para la publicación de una aplicación elemental.





9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral- profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

- Aplicar una evaluación diagnóstica al estudiante en base a conocimientos previos de análisis de sistemas, diseño de bases de datos, sistemas integrales de información en la arquitectura cliente-servidor.
- Para la evaluación formativa se aplicarán exámenes en cada unidad, tomando en cuenta participación en clase, realización de ejercicios, prácticas, trabajos y desarrollo del proyecto integrador entre otros, estableciendo criterios de evaluación y puntuaciones respectivas. En la evaluación sumativa se integran los resultados acumulados en cada actividad propuesta así como la puntualidad y oportunidad de la entrega correspondiente.





11. Fuentes de información

1. **Arias, A.** (2022). *Aprende a programar Ajax y JQuery*. IT Campus Academy.
2. **Blanc, L., Gouleau, E., Mansour, O., Rivoallan, T., Lemaire, V., & Lacot, X.** (2019). A comparative study of laravel and symfony PHP frameworks.
3. **Das, R., & Saikia, D. P.** (2016). Comparison of Procedural PHP with Codeigniter and Laravel Framework.
4. **Dwarampudi, V., Dhillon, S. S., Shah, J., Sebastian, N. J., & Kanigicharla, N. S.** (2019). Comparative study of the Pros and Cons of Programming languages Java, Scala, C++, Haskell, VB .NET, AspectJ, Perl, Ruby, PHP & Scheme Revision 1.0.
5. **Fayyaz, A., & Madiha, M.** (2014). Performance Evaluation of PHP Frameworks (CakePHP and CodeIgniter) in relation to the Object-Relational Mapping, with respect to Load Testing.
6. **Flórez, H., & Hernández, J.** (2021). *Aplicaciones Web con PHP*. Ediciones de la U.
7. **Laaziri, M., et al.** (2019). Procedia Manufacturing 32, 864–871. *A Guide to Popular PHP Frameworks for Beginners*.
8. **Marcano, A.** (2020). *Microsoft Asp.net MVC con Net Core 3.1*. Independently Published.
9. **Olanrewaju, R. F., Islam, T., & Ali, N.** (2022). An empirical study of the evolution of PHP MVC framework.
10. **Omelak, A.** (2020). *Laminas: MVC Framework for PHP*. Amazon Digital Services LLC - KDP Print US.
11. **Prokofyeva, N., & Boltunova, V.** (2016). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems.
12. **Pitt, C.** (2021). *Pro PHP 8 MVC: Model View Controller Architecture-Driven*. Apress.
13. **Richeton, N., & Version, C.** (2019). *Frameworks PHP*.
14. **Semeteys, R.** (2008). *Method for Qualification and Selection of Open Source Software*. Technology Innovation Management Review.
15. **Traversy, B.** (2018). *Object Oriented PHP and MVC*. Packt Publishing.

