

MÓDULO DE ESPECIALIDAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

Módulo de especialidad para el plan ISIC-2010-224 de la carrera de Ingeniería en Sistemas Computacionales, que tiene como objetivo proporcionar las competencias necesarias para que el (la) alumno (a) sea capaz de desarrollar aplicaciones inteligentes en la web y las tecnologías móvil.

*Desarrollo De
Aplicaciones
Inteligentes
Web-Móvil.*



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE JIQUILPAN

NOMBRE DE LA ESPECIALIDAD: **DESARROLLO DE APLICACIONES
INTELIGENTES WEB-MÓVIL.**

CLAVE DE LA ESPECIALIDAD: **ISIE-DAI-2024-03.**

PARA LA CARRERA: **INGENIERÍA EN SISTEMAS
COMPUTACIONALES.**

CLAVE DEL PLAN: **ISIC-2010-224**

FECHA DE INICIO: **ENERO 2024**

VIGENCIA: **3 AÑOS**

Jiquilpan Michoacán, 08 de mayo de 2023





ESPECIALIDAD: DESARROLLO DE APLICACIONES INTELIGENTES WEB-MÓVIL.

CLAVE DE LA ESPECIALIDAD: ISIE-DAI-2024-03.

FECHA DE INICIO: ENERO 2024

VIGENCIA: 3 AÑOS

OBJETIVO:

Al finalizar el módulo el (la) alumno(a) será capaz de desplegar aplicaciones híbridas inteligentes web-móvil que utilicen una arquitectura de microservicios y ofrezcan una mejor experiencia de usuario.

PERFIL DE LA ESPECIALIDAD:

El (la) Ingeniero(a) en Sistemas Computacionales tendrá la misión de detectar y satisfacer las necesidades organizacionales relacionadas al diseño de interfaz y experiencia de usuario, la implementación de algoritmos inteligentes para mejorar la experiencia de usuario, implementar una arquitectura de microservicios para tener aplicaciones con servicios distribuidos en la nube, diseño y despliegue de aplicaciones web y móvil. Desarrollará la capacidad para recabar y organizar los datos de los procesos necesarios en base a las tecnologías informáticas para el buen funcionamiento de la organización en el cumplimiento de sus objetivos.





El (la) Ingeniero(a) diseñará propuestas digitales de valor desde la fase de investigación hasta la creación del prototipo de baja y alta fidelidad en base a las necesidades del cliente y el usuario a través de técnicas de UX/UI.

Tendrá una preparación integral en el campo teórico y práctico, en áreas de Inteligencia Artificial, específicamente en los campos de visión por computadora y aprendizaje automático. Aplicará técnicas y algoritmos para crear un proyecto de Machine Learning en entornos offline y online. Evaluará modelos de redes neuronales convolucionales para problemas de visión por computadora.

El módulo de especialidad le ofrece la oportunidad de desplegar aplicaciones en microsistemas locales y en la nube facilitando su escalamiento, aprenderá buenas prácticas y configuración de un clúster Docker.

ASIGNATURAS DE LA ESPECIALIDAD:

No.	Asignatura	Créditos	Clave
1	Diseño de Interfaz y Experiencia de Usuario.	2-2-4	DAC-2401
2	Arquitectura Orientada a Servicios.	2-3-5	DAD-2402
3	Visión por Computadora y Machine Learning.	2-4-6	DAM-2403
4	Desarrollo de Aplicaciones Web.	2-3-5	DAD-2404
5	Desarrollo de Aplicaciones Móviles.	2-3-5	DAD-2405





COMPETENCIAS ESPECÍFICAS Y GENÉRICAS DE LAS ASIGNATURAS:

El módulo de especialidad de Desarrollo de Aplicaciones Inteligentes Web-Móvil, aporta al perfil de egreso las siguientes competencias específicas:

- Desarrolla un prototipado rápido y metodológico para el diseño de experiencia de usuario de una página web o aplicación móvil.
- Genera un diseño funcional y agradable para el usuario que permita hacer lo que desea de manera más intuitiva y sencilla posible.
- Comprende las distintas clases de interfaces del usuario.
- Desarrollar habilidades en el diseño, aplicación, análisis y diagnóstico de tests de usabilidad, en el diseño de interfaces de usuario logrando el prototipo.
- Identificar los elementos requeridos para la aplicación de un servicio web.
- Analizar las diferentes arquitecturas de servicios.
- Reconoce e identifica los diferentes tipos de servicios que se pueden desarrollar.
- Conocer el proceso que se requiere para la implementación de un servicio rest.
- Identificar las diferencias de las instrucciones de un lenguaje de microservicios con referencia a otros lenguajes.
- Conocer y aplicar los elementos e instrucciones del lenguaje a la solución de problemas reales.
- Aplicar los conceptos de clases y objetos en el desarrollo de programas para la solución de problemas de acuerdo al paradigma orientado a objetos.
- Utiliza técnicas del diseño para la solución de problemas con microservicios controlando sus excepciones.
- Aplicar la sintaxis de un lenguaje para crear microservicios que interactúen con otros microservicios.





- Realizar solución de problemas que integren microservicios desarrollados por otros programadores.
- Utiliza técnicas de diseño para la solución de problemas.
- Aplicar la sintaxis de un lenguaje para garantizar la seguridad e integridad del microservicio.
- Realizar solución de problemas de seguridad en la implementación de microservicios.
- Gestionar un servicio en la nube para el almacenamiento de microservicios disponibles para otros programadores.
- Implementar controles de acceso y uso de los microservicios desarrollados.
- Aplicar tecnologías que permitan el acceso a microservicios de forma remota.
- Proporcionar habilidades prácticas en programación, resolución de problemas, manipulación de datos numéricos, análisis de datos y optimización del rendimiento.
- Adquirir habilidades esenciales en visualización de datos, análisis exploratorio, comunicación efectiva y presentación de resultados.
- Tener habilidades en manipulación de datos, preparación de datos y trabajo con datos estructurados.
- Proporcionar una base sólida para trabajar en proyectos de aprendizaje profundo y aprovechar las oportunidades en campos como la inteligencia artificial, la visión por computadora y el procesamiento de lenguaje natural.
- Desarrollar habilidades y conocimientos específicos en el diseño, implementación y aplicación de redes neuronales convolucionales en el campo del aprendizaje automático y la visión por computadora.
- Desarrollar habilidades sólidas en el uso de redes neuronales recurrentes para el procesamiento del lenguaje natural, la traducción automática, la generación de texto.





- Comprende, investiga y selecciona el marco de trabajo más adecuados para desarrollar el front end y el back end.
- Habilidad para aplicar las herramientas gráficas del marco de trabajo en el diseño de una interfaz GUI.
- Capacidad para evaluar el diseño de la GUI aplicando UX.
- Capacidad para implementar la persistencia de datos en la web.
- Implementar un plan de seguridad en el proyecto web.
- Integración de servicios en el proyecto web.
- Capacidad para configurar el despliegue del desarrollo web en el servidor seleccionado.
- Capacidad para poner en producción un desarrollo web.
- Identificar los elementos requeridos en el desarrollo móvil.
- Analizar las diferentes alternativas de desarrollo móvil y requisitos necesarios.
- Reconoce e identifica los diferentes tipos de aplicaciones que se pueden desarrollar.
- Conocer el proceso que se requiere para publicar una aplicación.
- Identificar las diferencias de las instrucciones del lenguaje de programación móvil, con respecto a otros lenguajes de programación.
- Conocer y aplicar los elementos e instrucciones del lenguaje a la solución de problemas reales.
- Aplicar los conceptos de clases y objetos en el desarrollo de programas para la solución de problemas de acuerdo al paradigma orientado a objetos.
- Utiliza técnicas de modelado para la solución de problemas.
- Aplicar la sintaxis de un lenguaje para realizar interfaces de usuario utilizando componentes.
- Realizar solución de problemas para dispositivos móviles que integren manejo de eventos e interfaces adaptables a diferentes dispositivos.
- Utiliza técnicas de modelado para la solución de problemas.





- Aplicar la sintaxis de un lenguaje para realizar aplicaciones que utilicen sensores.
- Realizar solución de problemas para dispositivos móviles donde se apliquen elementos web, hilos o mapas.
- Desarrollar aplicaciones que utilicen el manejo de datos, considerando su entorno operativo.
- Implementar soluciones sobre dispositivos móviles que realicen manipulación de datos utilizando una base de datos.
- Aplicar tecnologías de conectividad de base de datos actuales y emergentes.
- Implementar la configuración del despliegue en el servidor seleccionado.
- Realizar la puesta en producción de la aplicación móvil.

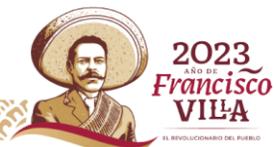
A su vez, este módulo de especialidad aporta las siguientes competencias genéricas al perfil del egresado:

Competencias instrumentales:

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Comunicación oral y escrita.
- Habilidad para buscar y analizar información proveniente de fuentes diversas.
- Solución de problemas.
- Toma de decisiones.
- Habilidades de gestión de información

Competencias interpersonales:

- Capacidad crítica y autocrítica.
- Trabajo en equipo.





- Capacidad de comunicación interdisciplinaria.
- Compromiso ético.
- Habilidades interpersonales.

Competencias sistémicas:

- Capacidad de aplicar los conocimientos en la práctica.
- Habilidades de investigación.
- Capacidad de aprender.
- Capacidad de generar nuevas ideas (creatividad).
- Capacidad de adaptarse a nuevas situaciones.
- Liderazgo.
- Habilidad para trabajar en forma autónoma.
- Búsqueda del logro.
- Capacidad para diseñar y gestionar proyectos.
- Iniciativa y espíritu emprendedor.
- Preocupación por la calidad.
- Conocimiento de culturas y costumbres de diversas organizaciones.

CONTENIDOS TEMÁTICOS DE LAS ASIGNATURAS:

Asignatura	Temas
Diseño de Interfaz y Experiencia de Usuario.	<ol style="list-style-type: none"> 1. Investigación y Prototipado. 2. Diseño de Interacción. 3. Diseño de Interfaces. 4. Analítica y Test con Usuarios.





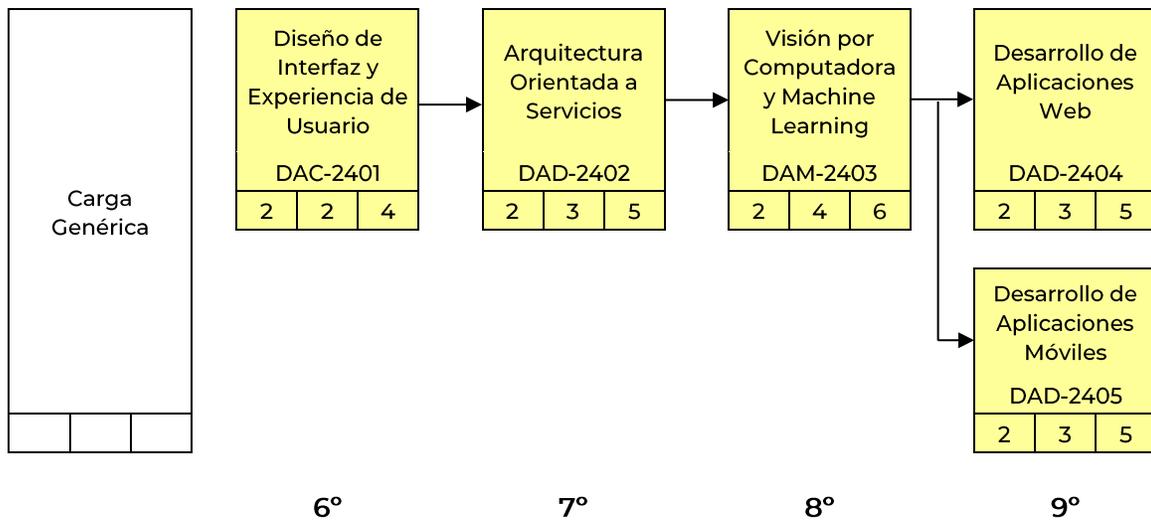
<p>Arquitectura Orientada a Servicios.</p>	<ol style="list-style-type: none"> 1. Fundamentos Sobre Servicios Web. 2. Desarrollo de Microservicios. 3. Interacción Entre Microservicios. 4. Seguridad en Microservicios. 5. Despliegue de Microservicios en la Nube.
<p>Visión por Computadora y Machine Learning.</p>	<ol style="list-style-type: none"> 1. Procesamiento y Análisis de Datos. 2. Fundamentos de Deep Learning. 3. Redes Neuronales Convolucionales. 4. Redes Neuronales Recurrentes.
<p>Desarrollo de Aplicaciones Web.</p>	<ol style="list-style-type: none"> 1. Marcos de Trabajo. 2. Interfaces Gráficas (GUI). 3. Persistencia de Datos. 4. Seguridad. 5. Servicios. 6. Despliegue.
<p>Desarrollo de Aplicaciones Móviles.</p>	<ol style="list-style-type: none"> 1. Introducción a la Programación Móvil. 2. Lenguaje de Programación Móvil. 3. Desarrollo de Aplicaciones Móviles. 4. Elementos Avanzados de Desarrollo Móvil. 5. Administración de Datos en Dispositivos Móviles. 6. Despliegue de Aplicaciones Móviles.



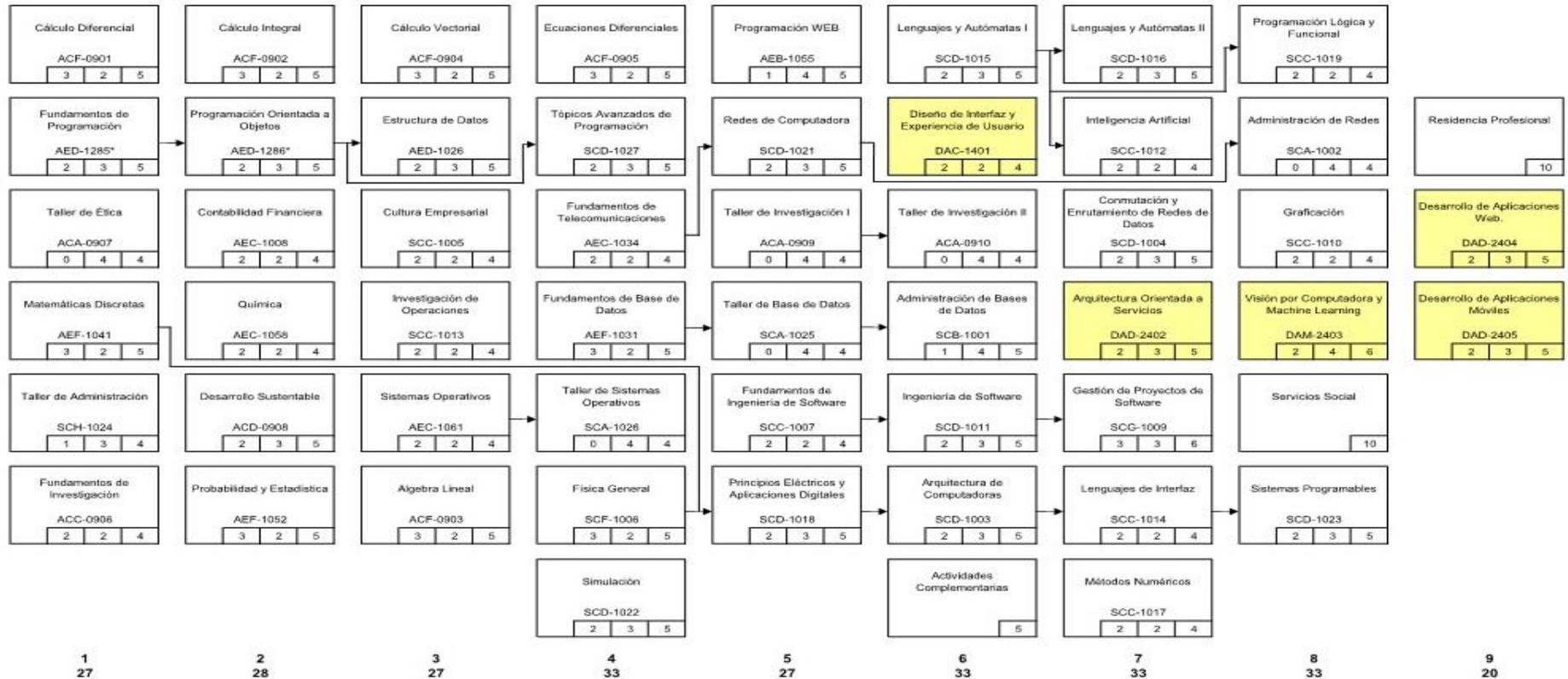


MAPA CURRICULAR DEL MÓDULO:

Las asignaturas del módulo de especialidad se ofrecen a partir del sexto semestre y concluyen en el noveno como se indica a continuación:



Ingeniería en Sistemas Computacionales
ISIC-2010-224
ISIE-DAI-2024-03



Genéricas	211
Residencias	10
Servicio Social	10
Otros	5
Especialidad	25
Total de Créditos	261



1. Datos Generales de la asignatura

Nombre de la asignatura:	Diseño de Interfaz y Experiencia de Usuario.
Clave de la asignatura:	DAC-2401
SATCA¹:	2-2-4
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura
<p>Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las competencias profesionales:</p> <ul style="list-style-type: none"> • Aplicar conocimientos científicos y tecnológicos en la solución de problemas en el área informática con un enfoque interdisciplinario. • Administrar las tecnologías de la información, para estructurar proyectos estratégicos. • Analizar, modelar, desarrollar a través de técnicas de usuarios el prototipo que reflejará una aplicación funcional a cualquier organización. • La importancia de esta asignatura es que permite al estudiante llevar a cabo un análisis previo de funcionalidad.
Intención didáctica
<p>La asignatura debe ser práctica y capaz de desarrollar en el estudiante la habilidad de realizar un prototipo funcional para cualquier organización de una interfaz con experiencia de usuario</p> <ul style="list-style-type: none"> • El tema I. El estudiante será capaz de desarrollar un prototipado rápido y metodológico para el diseño de experiencia de usuario de una página web o aplicación móvil. • El tema II. El estudiante será capaz de generar un diseño funcional y agradable para el usuario que permita hacer lo que desea de manera más intuitiva y sencilla posible.

¹ Sistema de Asignación y Transferencia de Créditos Académicos





- El tema III. El estudiante comprende las distintas clases de interfaces del usuario.
- El tema IV. El estudiante será capaz de implementar en un sitio web o una app, los test necesarios para mejorar la experiencia de usuario.

3. Participantes en el diseño y seguimiento curricular del programa

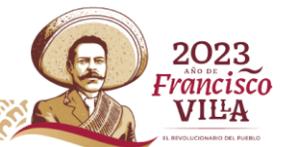
Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Jiquilpan Jiquilpan Michoacán, 08 de mayo de 2023	Ing. Dalet Villanueva Mascort Lic. Martha Elizabeth González Zepeda M.C. Edna Paola Chávez Guerra M.C. María Elena Cabrera Rocha Ing. Fernando Carranza Campos	Diseño y desarrollo de los programas de estudio del módulo de especialidad para la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<ul style="list-style-type: none"> • Lograr establecer estrategias, a través de un marco de acción basado en la experiencia del usuario, obtenidas desde sus necesidades, logrando con esto mejorar el desarrollo analítico del estudiante.

5. Competencias previas

<p>Esta asignatura, es la aplicación práctica del conocimiento científico, a través de los métodos y técnicas adecuados, para el desarrollo de aplicaciones web y móvil.</p> <p>Tiene relación previa con las signaturas asignaturas:</p> <ul style="list-style-type: none"> • Taller de Administración. • Fundamentos de Ingeniería de Software.





- Programación Web.

En base a las asignaturas previas, se consideran las siguientes competencias específicas:

- Aplicar principios de la administración.
- Aplicar principios de ingeniería del software en las etapas de análisis y diseño de un sistema de información.
- Prototipado del proyecto.

Desarrollar propuestas con base en los elementos teórico-práctico adquiridos que permitan generar un prototipo funcional de acuerdo a las necesidades del usuario.

6. Temario

No.	Temas	Subtemas
1	Investigación y Prototipado.	1.1. Introducción a la investigación y el prototipado rápido. 1.2. Técnicas de investigación cualitativas y cuantitativas. 1.2.1. Técnicas de investigación cualitativa. 1.2.2. Técnicas de investigación cuantitativa. 1.2.3. Análisis de resultados de la investigación. 1.3. Personas mapas y empatía. 1.4. Customer Journey. 1.5. Necesidades del usuario. 1.6. Prototipado rápido.
2	Diseño de Interacción.	2.1. Introducción al diseño de interacción. 2.2. Arquitectura de Información. 2.3. Flujos de navegación.





		2.4. Wireframing.
3	Diseño de Interfaces.	3.1. Introducción al diseño de interfaces. 3.2. Fundamentos del diseño de interfaces. 3.3. Branding. 3.4. Sistemas de diseño.
4	Analítica y Test con Usuarios.	4.1. Introducción al test con usuarios. 4.2. Implementación del test con usuarios.

7. Actividades de aprendizaje de los temas

I Investigación y Prototipado.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Desarrolla un prototipado rápido y metodológico para el diseño de experiencia de usuario de una página web o aplicación móvil. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Habilidad para buscar y analizar información proveniente de fuentes diversas. Comunicación oral y escrita. 	<ul style="list-style-type: none"> Realizar un mapa mental de las técnicas para elaborar un prototipado rápido. Investigar y discutir en equipos los tipos de técnicas de investigación. Realizar un documento que ilustre visualmente la relación del cliente con la empresa y su percepción de la misma, así como sus necesidades y expectativas en cada fase de la relación.
II Diseño de interacción.	



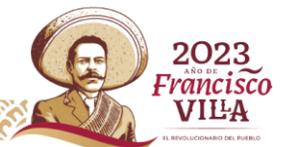


Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Genera un diseño funcional y agradable para el usuario que permita hacer lo que desea de manera más intuitiva y sencilla posible. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Capacidad de aplicar los conocimientos en la práctica. • Toma de decisiones. 	<ul style="list-style-type: none"> • Investigar las diferentes técnicas del diseño de interacción para realizar un cuadro comparativo. • Realizar el diseño funcional a través de una plantilla de Wireframing para mostrar los elementos de la interface.
III Diseño de Interfaces.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Comprende las distintas clases de interfaces del usuario. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. 	<ul style="list-style-type: none"> • Investigar las diferentes herramientas del diseño de interfaces. • Realizar un cuadro comparativo de diferentes herramientas del diseño de interfaces. • Realizar por medio de la herramienta de diseño la gestión de marca. • Implementar el proyecto a través del diseño de interfaz y la gestión de marca.





<ul style="list-style-type: none"> Habilidad para buscar y analizar información proveniente de fuentes diversas. Solución de problemas. Toma de decisiones. Capacidad crítica y autocrítica. Trabajo en equipo. Habilidades interpersonales. Capacidad de aplicar los conocimientos en la práctica. 	
<p>IV Analítica y Test con Usuarios.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <ul style="list-style-type: none"> Desarrollar habilidades en el diseño, aplicación, análisis y diagnóstico de tests de usabilidad, en el diseño de interfaces de usuario logrando el prototipo. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Habilidad para buscar y analizar información proveniente de fuentes diversas. Solución de problemas. Toma de decisiones. Capacidad crítica y autocrítica. Trabajo en equipo. Habilidades interpersonales. 	<ul style="list-style-type: none"> Analizar los tipos de perfiles de usuarios para determinar la muestra. Identificar el tipo de perfil de usuario a utilizar en el prototipo. Investigar el tipo de método que se va a emplear en la etapa de test que se aplicará al usuario. Realizar la prueba piloto. Realizar el prototipado del proyecto integrador.





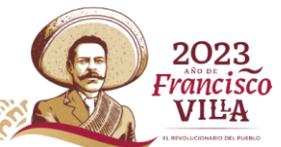
<ul style="list-style-type: none"> • Capacidad de aplicar los conocimientos en la práctica. 	
--	--

8. Práctica(s)

<p>Tema I</p> <p>Práctica 1. Analizar los resultados de la obtención de la recolección de datos de acuerdo a los mapas de empatía de las personas.</p> <p>Práctica 2. Realizar el diseño de un prototipo de baja fidelidad de acuerdo con las necesidades de usuario y el análisis de datos.</p> <p>Tema II</p> <p>Práctica 1. Presentar un modelo de arquitectura de información de acuerdo con el diseño de interacción.</p> <p>Práctica 2. Realizar un modelo de flujo de información para interpretación en el wireframing.</p> <p>Práctica 3. Interpolación del modelo de arquitectura y el modelo de flujo de información en un wireframing.</p> <p>Tema III</p> <p>Práctica 1. Seleccionar el enfoque del proyecto para realizar el diseño de interfaz y gestión de marca.</p> <p>Tema IV</p> <p>Práctica 1. Analizar el sondeo de la información resultante para determinar el perfil del usuario.</p> <p>Práctica 2. Implementar el mejor método para el prototipado del proyecto.</p> <p>Práctica 3. Presentar los resultados de los test implementados en el prototipado del proyecto.</p>

9. Proyecto de asignatura

Fundamento:





El objetivo del proyecto es desarrollar en equipo una propuesta de un prototipo para una aplicación que dé solución a un problema real. Los proyectos a realizar se establecen a partir de especificaciones que realice el docente o necesidades que los estudiantes determinen que existen en el entorno.

El prototipo de la aplicación ayudará a visualizar de una manera más efectiva la apariencia que se desea sea con la cual interactúe el usuario final, ya que crear una experiencia de usuario es un proceso clave para conseguir el éxito de la organización.

Planeación:

Los equipos serán de tres integrantes como máximo, el proyecto a realizar podrá considerar en su desarrollo las competencias adquiridas en las asignaturas relacionadas con el desarrollo de software; así como los principios del desarrollo ágil y la experiencia de usuario.

Desarrollo:

Durante el desarrollo del prototipo se llevarán a cabo las actividades planteadas en la planeación y a las necesidades del proyecto, pero se desarrollan de forma particular algunas propuestas que deben de cumplirse.

La aplicación debe contemplar:

- Hacer un prototipo que genere amigabilidad y cercanía del producto con el usuario.
- El diseño del prototipo deberá ser acorde a las políticas de la organización que recibirá el beneficio.
- Que los diferentes integrantes en el desarrollo del prototipo participen en él.
- Diseño del prototipado del proyecto integrador.
- Documentación del prototipo.

Evaluación

La rúbrica contemplará los siguientes criterios:

Planeación 30%

Desarrollo 40%

Exposición del proyecto 20%

Trabajo en equipo 10%

10. Evaluación por competencias





La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño encada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Rúbricas o productos, señalados en cada unidad académica dentro de las actividades de aprendizaje.
- Prácticas propuestas y su presentación y exposición en plenaria. Algunas se evaluarán por equipo.
- Portafolio de evidencias de Información recabada durante las consultas e investigaciones solicitadas, plasmadas en documentos escritos.
- Descripción de otras experiencias concretas que se obtendrán al participar en discusiones, exposiciones o cualquier otro medio didáctico-profesional que trate sobre la materia y que deberán realizarse durante el curso académico.
- Exámenes teórico-prácticos para comprobar la efectividad del estudiante en la resolución de casos prácticos.

11. Fuentes de información

Tidwell, Jennifer (2006). Designing Interfaces - Patterns for Effective Interaction Design (Ed. O'REILLY).

Jorquera, F. y Zamora, A. (05 de junio de 2020) Microcopy para una mejor UX: ¿Cómo las palabras nos ayudan a tomar decisiones?

Zamora, A. (2020) UX Writing: Creando experiencias de usuario desde la comunicación en Blog IDA.

Fenton, N. (2014). Nicely Said: Writing for the Web with Style and Purpose. Berkley: New Riders. Gillen, P. Lead with Content: How to put content at the centre of digital transformation. Brighton: Gather Content Limited.

Lichaw, D. (2016). The User's Journey: Storymapping Products That People Love. Nueva York: Rosenfeld Media LLC.

Lupton, E. (2019). El diseño como Storytelling. (Edición en español) Barcelona: Editorial Gustavo Gili.

Metts, J. y Welfle, A. (2020). Writing is designing: Words and the User Experience. Nueva York: Rosenfeld Media LLC.

Referencias:





http://www.greenonions.com/portfolio/dbrown_ia2005_wireframes.pdf
http://www.boxesandarrows.com/view/where_the_wireframes_are_special_deliverable
<http://www.jjg.net/ia/visvocab/spanish.html>
http://www.guuui.com/issues/02_03_02.php
http://www.boxesandarrows.com/view/storyboarding_rich_internet_applications_with_visio
<http://www.infomed.sld.cu>
<http://www.ayerviernes.com>
<http://www.nosolousabilidad.com/articulos/diagramacion.htm>
<http://www.human-computer.net/blog/2012/10/stencils-para-hacer-wireframes-con-microsoft-visio/>
<http://olgacarreras.blogspot.com.es/2007/02/wireframes.html>
<http://www.creativebloq.com/wireframes/top-wireframing-tools-11121302>





1. Datos Generales de la asignatura

Nombre de la asignatura:	Arquitectura Orientada a Servicios.
Clave de la asignatura:	DAD-2402
SATCA¹:	2-3-5
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las competencias profesionales:

- Tener un entendimiento profundo de los principios y conceptos arquitectónicos en el contexto de SOA y microservicios. Esto incluye comprender cómo diseñar y construir servicios independientes, cómo establecer comunicación entre ellos y cómo crear una arquitectura modular y escalable.
- Diseñar interfaces de servicios y a definir protocolos que permitan la interoperabilidad y la comunicación efectiva entre los diferentes componentes de un sistema distribuido. También desarrollar habilidades en el modelado de servicios, la identificación de límites de contexto y la definición de responsabilidades claras para cada microservicio.
- Obtener habilidades en la implementación de servicios utilizando tecnologías y frameworks específicos, como SOAP, APIs RESTful/JSON, Flask, Django REST, Framework, Node.js, etc. desarrollar microservicios que sean independientes y reutilizables, centrándote en la modularidad, la cohesión y el acoplamiento bajo.
- Gestionar la comunicación y la integración entre los diferentes servicios y componentes de un sistema. Esto incluye el uso de protocolos de comunicación, como HTTP o eventos, y la comprensión de patrones y herramientas para el enrutamiento, la

¹ Sistema de Asignación y Transferencia de Créditos Académicos





autenticación, la seguridad y la monitorización de las interacciones entre servicios.

- Desarrollar habilidades para identificar y resolver problemas en sistemas distribuidos, como errores de comunicación, problemas de rendimiento o fallas en la integración entre servicios.

Estas competencias profesionales proporcionarán una base sólida para diseñar, desarrollar y mantener sistemas basados en arquitectura orientada a servicios (SOA) y microservicios. Además, también brindarán una comprensión más amplia de los principios de diseño de software y arquitectura, lo que puede ser aplicable a otros enfoques arquitectónicos y tecnologías.

Intención didáctica

La asignatura debe ser práctica y capaz de despertar en el estudiante la habilidad para desarrollar microservicios que se puedan implementar en la web, dispositivos móviles y aplicaciones de escritorio, además cubre con la necesidad inminente que tiene un ingeniero al enfrentarse al uso cotidiano de tecnologías.

El temario está organizado en cinco temas. El primer tema, introduce al estudiante a los conceptos de transferencia de estado representacional o servicios REST.

El tema dos, se centra en el aprendizaje del funcionamiento de los microservicios, desde su configuración hasta su empaquetado, pasando por el envío y acceso a datos.

El tema tres, permite la comprensión de como interactúan los microservicios, la manipulación de sus respuestas y el control de sus posibles excepciones.

En el tema cuatro, el alumno aplica las herramientas de seguridad de los microservicios garantizando la fiabilidad de sus resultados y generar la confianza en el microservicio desarrollado.

Por último, el tema cinco se enfoca en el despliegue de los microservicios en la nube, facilitando su uso en proyecto diversos de la ingeniería en el desarrollo de software.





3. Participantes en el diseño y seguimiento curricular del programa

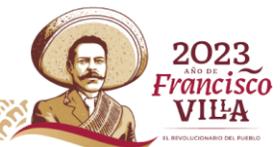
Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Jiquilpan Jiquilpan Michoacán, 08 de mayo de 2023	Ing. Leonardo Martínez González Ing. Erik Dalet Villanueva Mascort Lic. Martha Elizabeth González Zepeda Lic. Armando Hernández Gálvez	Diseño y desarrollo de los programas de estudio del módulo de especialidad para la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<ul style="list-style-type: none"> • Identifica los elementos que intervienen en el desarrollo de microservicios. • Utiliza y maneja entornos de desarrollo para la creación de microservicios. • Utiliza técnicas de modelado para la solución de problemas empleado los microservicios. • Aplica un lenguaje de programación para la solución de problemas que requieran uso de microservicios. • Aplica algoritmos de inteligencia artificial en la solución de problemas.

5. Competencias previas

<ul style="list-style-type: none"> • Analiza y soluciona problemas informáticos y representa su solución mediante herramientas de software orientado a objetos. • Identifica y analiza necesidades de información para su representación, tratamiento y automatización para la toma de decisiones. • Diseña esquemas de base de datos para generar soluciones al tratamiento de información. • Identifica y aplica distintos SGBD, así como sus herramientas. • Aplica el entorno operativo del SGBD, para crear el esquema de una base de datos. • Aplica esquema de integridad, seguridad y recuperación.





6. Temario

No.	Temas	Subtemas
1	Fundamentos Sobre Servicios Web.	1.1. Servicios REST. 1.2. Servicios web y arquitectura de servicios. 1.3. Análisis de un servicio REST.
2	Desarrollo de Microservicios.	2.1. Propiedades de configuración. 2.2. Envío de datos desde el cliente. 2.3. Mapeo de objetos. 2.4. Respuesta de diferentes métodos http. 2.5. Testing de microservicios. 2.6. Acceso a datos desde un microservicio. 2.7. Empaquetado y despliegue de un microservicio.
3	Interacción Entre Microservicios.	3.1. Interacción entre microservicios. 3.2. Interacción en modo asíncrono. 3.3. Manipulación de respuestas. 3.4. Excepciones en servicios REST. 3.5. Generación de respuestas de servicios.
4	Seguridad en Microservicios.	4.1. Securitización del acceso a un microservicio. 4.2. Acceso a servicio securizado. 4.3. Token JWT. 4.4. Securitización de microservicios con token JWT. 4.5. El servidor de identidades keycloak. 4.6. Configuración de usuarios en keycloak.

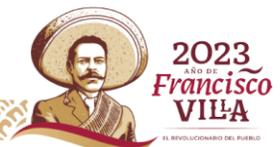




5	Despliegue de Microservicios en la Nube.	<ul style="list-style-type: none"> 5.1. Soluciones cloud. 5.2. Registro de microservicios. 5.3. Securitización acceso. 5.4. Cliente. 5.5. Acceso remoto a microservicios.
---	--	--

7. Actividades de aprendizaje de los temas

I Fundamentos Sobre Servicios Web.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Identificar los elementos requeridos para la aplicación de un servicio web. • Analizar las diferentes arquitecturas de servicios. • Reconoce e identifica los diferentes tipos de servicios que se pueden desarrollar. • Conocer el proceso que se requiere para la implementación de un servicio REST. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Comunicación oral y escrita. 	<ul style="list-style-type: none"> • Realizar una investigación de la arquitectura de microservicios. • Estudiar casos reales de implementación de microservicios desde cero, identifica los límites del contexto, diseño de interfaz y políticas de acceso. • Hacer una tabla comparativa de las diferentes tecnologías en el desarrollo de servicios REST. • Realizar un comparativo de ventajas y desventajas de los diferentes tipos de servicios. • Realizar un diagrama de flujo de procesos, la secuencia de actividades que se necesita para obtener un servicio REST.





II Desarrollo de Microservicios.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Implementar microservicios utilizando principios arquitectónicos de microservicios, identificando y dividiendo adecuadamente un sistema de servicios autónomos y coherentes, aplicar protocolos de comunicación como HTTP, REST, SOAP y gestionar la disponibilidad de los servicios en entornos de producción. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Capacidad de aplicar los conocimientos en la práctica. Capacidad crítica y autocrítica. Trabajo en equipo. Toma de decisiones. 	<ul style="list-style-type: none"> Investigar y analizar la arquitectura de microservicios de Netflix, analizar los retos de escalabilidad, tolerancia a fallos y rendimientos de la plataforma. (https://netflixtechblog.com/). Diseñar un sistema de microservicios, elegir un sistema o una aplicación y diseñar su arquitectura de microservicios. Identifica los límites de contexto, define los servicios y establece las interfaces y las políticas. Considera los principios de diseño y las mejores prácticas para garantizar una arquitectura escalable y flexible. Implementar el microservicio, utiliza un lenguaje de programación y un framework adecuado para desarrollar los microservicios. Utilizar protocolos de comunicación como HTTP, REST y/o SOAP para la interacción entre los servicios. Desplegar los microservicios en entornos de producción, considerar aspectos como la orquestación de contenedores, la escalabilidad horizontal y la gestión de la infraestructura en la nube. Utiliza plataformas como Kubernetes, Docker u otras soluciones de implementación.
III Interacción Entre Microservicios.	
Competencias	Actividades de aprendizaje





<p>Específica(s):</p> <ul style="list-style-type: none"> comprender y aplicar técnicas efectivas para la interacción entre microservicios, asegurando una comunicación eficiente, confiable y escalable dentro de un sistema distribuido. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Habilidad para buscar y analizar información proveniente de fuentes diversas. Solución de problemas. Toma de decisiones. Capacidad crítica y autocrítica. Trabajo en equipo. Habilidades interpersonales. Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> Estudiar y comprender diferentes patrones y prácticas para la comunicación efectiva entre microservicios, como el patrón de mensajería, eventos, API Gateway, sincronización y coordinación de servicios, y otros patrones comunes en arquitecturas de microservicios. Elegir un caso de uso práctico e implementar la comunicación entre microservicios utilizando un lenguaje de programación y un framework de tu elección. Comenzar con un caso de uso sencillo y avanzar gradualmente hacia casos más complejos. Estudiar sistemas existentes basados en microservicios y analizar cómo se han implementado las interacciones entre los diferentes microservicios. Estudiar la arquitectura, los contratos de API, los mecanismos de comunicación utilizados y los patrones de manejo de errores y tolerancia a fallos implementados. Intenta comprender las decisiones de diseño detrás de esas implementaciones y extrae lecciones de ellas. Participar en proyectos colaborativos de código abierto o equipos de desarrollo donde se implementen arquitecturas de microservicios. Experimentar con herramientas y tecnologías como: soluciones de mensajería y cola de eventos, sistemas de gestión de API, herramientas de monitorización y seguimiento, y plataformas de enrutamiento y balanceo de carga. Probar diferentes enfoques y evaluar cómo afectan
--	--





	la comunicación y el rendimiento de los microservicios.
IV Seguridad en Microservicios.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Garantizar la seguridad de los microservicios y mitigar los riesgos de seguridad en un entorno distribuido. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Habilidad para buscar y analizar información proveniente de fuentes diversas. Solución de problemas. Toma de decisiones. Capacidad crítica y autocrítica. Trabajo en equipo. Habilidades interpersonales. Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> Desarrolla e implementa servicios seguros que requieran autenticación en su acceso. Desarrolla e implementa servicios que utilicen Token JWT, garantizando su seguridad. Desarrolla, implementa y configura servicios de identidades keycloak. Implementar la encriptación y protección de datos mediante algoritmos criptográficos adecuados para proteger la confidencialidad y la integridad de los datos en tránsito y en reposo. Participar en comunidades de seguridad y seguir las mejores prácticas de seguridad en el desarrollo, implementación y gestión de los microservicios.
V Despliegue de Microservicios en la Nube.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p>	<ul style="list-style-type: none"> Realizar un estudio de las plataformas de nube, comprender las plataformas de nube





<ul style="list-style-type: none"> • Desplegar eficientemente microservicios en entornos de computación en la nube, aprovechando las ventajas de escalabilidad, disponibilidad y flexibilidad que ofrecen estas plataformas. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. • Capacidad de aplicar los conocimientos en la práctica. 	<p>disponibles, como AWS, Azure o Google Cloud Platform. Aprender sobre los servicios ofrecidos por cada plataforma para la gestión de microservicios, características, precios y capacidades de escalabilidad y disponibilidad.</p> <ul style="list-style-type: none"> • Configurar y administrar clústeres de contenedores en un entorno de desarrollo local o utilizando servicios de nube para comprender cómo funcionan y cómo gestionar los microservicios. • Diseñar y configurar la escalabilidad y disponibilidad de microservicios en la nube. Practicar la configuración de políticas de escalado automático, la distribución de carga y la configuración de grupos de autoescalado para asegurar que los microservicios puedan manejar cargas de trabajo cambiantes y sean altamente disponibles. • Utilizar herramientas nativas de la plataforma de nube para recopilar métricas y registros de los microservicios. Establecer alertas y practicar la visualización de datos para comprender el rendimiento y la salud de los microservicios. • Aprender a gestionar eficientemente los costos y optimizar el rendimiento de los microservicios en la nube.
---	---

8. Práctica(s)

<p>Tema I</p> <p>Práctica 1. Pasos para implementar un microservicio desde cero (aplicación de comercio electrónico).</p>





1. Analizar los requerimientos y las funcionalidades de la aplicación para identificar los límites del contexto y definir las responsabilidades de cada microservicio. Divide la funcionalidad en dominios más pequeños y coherentes, asignando cada dominio a un microservicio específico.
2. Diseñar las interfaces/APIs que expondrán las funcionalidades necesarias para cada microservicio. Define los endpoints, los métodos HTTP, los parámetros y las respuestas esperadas para cada operación. Estos serán parte del contrato del microservicio.
3. Configurar el entorno de desarrollo para cada microservicio. Esto puede incluir la instalación de frameworks o librerías necesarias, configuración de bases de datos, servicios de mensajería, etc. Asegurarse de tener un ambiente aislado para cada microservicio.
4. Implementar los microservicios por separado. Utiliza un lenguaje de programación y framework para desarrollar las funcionalidades requeridas en cada microservicio, seguir los principios y patrones de diseño de microservicios.
5. Realizar pruebas unitarias y pruebas de integración para cada microservicio para asegurarte de que funcionen correctamente y cumplan con los requisitos establecidos en el contrato.
6. Configurar el enrutamiento y la comunicación entre microservicios: Utiliza protocolos como HTTP, REST o mensajería asíncrona para que los microservicios se comuniquen entre sí.
7. Agregar aspectos de seguridad: Implementa mecanismos de seguridad en los microservicios, como autenticación y autorización, para proteger los datos y garantizar el acceso seguro a las funcionalidades. Utiliza estándares de seguridad como JWT o OAuth según corresponda.
8. Desplegar cada microservicio en un entorno de producción o en un entorno de pruebas, según sea necesario. Configura la infraestructura requerida para cada microservicio y asegúrate de que estén disponibles para su consumo.
9. Implementar herramientas de monitoreo para obtener información sobre el rendimiento de los microservicios.
10. Escalar y evolucionar a medida que la aplicación crezca y evolucione, considerar la posibilidad de agregar nuevos microservicios para abordar nuevas funcionalidades.

Tema II

Práctica 1. Descomposición de monolito a microservicios utilizando Python, Docker y MongoDB:

1. Analiza el monolito existente:





Estudia la estructura y el código del monolito para comprender su funcionamiento y las diferentes funcionalidades que abarca.

Identifica las áreas del monolito que pueden separarse en microservicios independientes.

2. Define los límites del contexto y responsabilidades:

Identifica las funcionalidades o dominios específicos que pueden convertirse en microservicios separados.

Define los límites del contexto de cada microservicio, asegurándote de que tenga una única responsabilidad claramente definida.

3. Diseña las interfaces de API:

Para cada microservicio, define las interfaces de API utilizando el framework web Flask de Python.

Define los endpoints y los métodos HTTP para cada funcionalidad del microservicio.

4. Configura la base de datos MongoDB:

Utiliza Docker para configurar un contenedor de MongoDB en tu entorno de desarrollo.

Define los esquemas de datos necesarios para cada microservicio y configura las conexiones correspondientes en cada uno.

5. Implementa los microservicios:

Crea un proyecto separado para cada microservicio utilizando Python y el framework Flask.

- Implementa la lógica de negocio de cada microservicio en su propio proyecto, utilizando las bibliotecas y los módulos necesarios.

6. Empaqueta los microservicios en contenedores Docker:

- Crea un archivo Dockerfile para cada microservicio, especificando las dependencias y configuraciones necesarias.

- Utiliza Docker para construir las imágenes de contenedor para cada microservicio.

- Asegúrate de configurar adecuadamente las variables de entorno y las conexiones a la base de datos MongoDB en cada contenedor.

7. Configura la comunicación entre microservicios:

- Utiliza HTTP y JSON como protocolos de comunicación para permitir que los microservicios se comuniquen entre sí.





- Define las llamadas HTTP y los formatos de datos para intercambiar información entre los microservicios.

8. Realiza pruebas y validaciones:

- Implementa pruebas unitarias y de integración para cada microservicio, utilizando herramientas como pytest o unittest.
- Verifica que cada microservicio funcione correctamente y se comuniquen adecuadamente con los demás.

9. Despliega los microservicios:

- Utiliza Docker y herramientas de orquestación de contenedores, como Docker Compose o Kubernetes, para desplegar los microservicios en un entorno de producción.
- Configura la escalabilidad y el balanceo de carga según sea necesario.

10. Monitorea y mantén los microservicios:

- Implementa herramientas de monitorización, como Prometheus o ELK Stack, para supervisar el rendimiento y la salud de los microservicios.
- Configura alertas y notificaciones para detectar y responder a eventos y anomalías.
- Realiza ajustes y actualizaciones según sea necesario para mantener la disponibilidad y el rendimiento de los microservicios.

Recuerda que este es solo un ejemplo de cómo abordar la descomposición de un monolito a microservicios utilizando Python, Docker y MongoDB. Puedes adaptar los pasos y las tecnologías según tus necesidades y preferencias específicas.

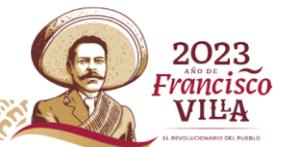
Tema III

Práctica 1. Crear una aplicación de notificaciones en Python que utilice MongoDB para almacenar datos y RabbitMQ para comunicarse con el servicio de envío de correos electrónicos de forma asíncrona.

En esta práctica, se crearán tres componentes: el servicio de "Notificaciones", el servicio de "Correo electrónico" y RabbitMQ como sistema de mensajería. Utilizar el módulo 'pika' para la comunicación con RabbitMQ y 'pymongo' para interactuar con MongoDB.

1. Configuración de RabbitMQ.

- Instalar RabbitMQ.





- Configurar RabbitMQ para envío de mensajes (notificaciones)
- Verificar que RabbitMQ esté instalado y en ejecución en el sistema.

2. Configuración de MongoDB

- Instalar MongoDB creando una cuenta en Atlas u otras opciones.
- Instalar MongoDB Compass en la computadora local para realizar la conexión y pruebas
- Crear una base de datos y una colección llamada para almacenar las notificaciones.

3. Crear el servicio de Notificaciones en python

- Importar las librerías necesarias: pika, pymongo, json, etc,
- Configurar y establecer la conexión a RabbitMQ
- Configurar y establecer la conexión a la Base de Datos de MongoDB y a la colección.
- Crear el método para procesar y enviar las notificaciones.
- Crear el método para recibir el evento de RabbitMQ y enviar notificación.

4. Crear el servicio de Correo electrónico en Python

- Importar las librerías: pika, json, etc.
- Configurar y establecer la conexión a RabbitMQ.
- Crear el método para enviar correos electrónicos.
- Crear el método para recibir el evento de RabbitMQ y enviar el correo electrónico.

5. Enviar notificaciones desde una aplicación externa

- Importar las librerías: pika, json, etc.
- Configurar y establecer la conexión a RabbitMQ.
- Crear un método para enviar la notificación a la cola de RabbitMQ
- Crear el programa principal para que llame al método anterior.

6. Ejecutar los servicios

- En una terminal, ejecuta el servicio de "Notificaciones".
- En otra terminal, ejecuta el servicio de "Correo electrónico".
- Ejecuta la aplicación externa para enviar una notificación.

7. Verificar los resultados

Práctica 2. Interacción basada en API entre dos microservicios utilizando Python y el framework Flask.





Se cuenta dos microservicios: un servicio de "Productos" y un servicio de "Carrito de compras". El servicio de "Productos" proporciona información sobre los productos disponibles y el servicio de "Carrito de compras" permite a los usuarios agregar productos a su carrito.

1. Configurar de Flask y crear los servicios

- Crear el servicio de productos, importar librerías, crear una lista con productos disponibles, crear la ruta para obtener los productos, definir el puerto, por ejemplo el 5000.
- Crear el servicio de Carrito de compras, importar librerías, crear la lista para simular el carrito, crear la ruta para obtener el contenido del carrito, crear la ruta para agregar un producto al carrito de compras, definir un puerto diferente al servicio anterior, por ejemplo 5001.

2. Ejecutar los servicios

- Ejecutar los servicios cada uno en una terminal diferente

3. Realizar las peticiones a las APIs

- Utilizar una herramienta como Postman o realizar peticiones HTTP desde Python para interactuar con las APIs de los microservicios.

4. Revisa los resultados

- Obtener los productos del servicio Productos
- Agregar un producto al carrito de compras
- Mostrar el contenido actual del carrito de compras.

Tema IV

Práctica 1. Crear un servicio de autenticación que emita tokens JWT para el inicio de sesión y la verificación del token. Implementar un API Gateway (Kong) para proteger los microservicios y asegurar de que solo los usuarios autenticados puedan acceder a ellos.

1. Configuración de Kong

- Instalar Kong y asegurarse de que esté en funcionamiento. Seguir la documentación oficial de Kong para la instalación: <https://docs.konghq.com/install/>





2. Crear el servicio de autenticación

- Importar las librerías: Flask, jwt, datettime, etc
- Crear una lista de usuarios registrados.
- Crear la ruta y el método para el login de usuario.
- Establecer el puerto http 5002.

3. Configurar el API Gateway Kong

- Asegurarse de que Kong esté en funcionamiento. Puedes seguir la documentación oficial de Kong para la configuración: <https://docs.konghq.com/getting-started/configuring-a-service/>
- Configurar el API Gateway Kong para enrutar las solicitudes entrantes al servicio de autenticación y asegúrate de que la autenticación se realice mediante JWT.

4. Crear los microservicios protegidos

- Importar las librerías: Flask, jwt, datettime, etc.
- Definir la palabra secreta
- Definir la lista de productos.
- Definir la ruta y verificar el token JWT antes de devolver los productos.
- Establecer el puerto HTTP 5000.

5. Realizar las peticiones a través del API Gateway

- Importar la librería: requests.
- Crear el método para el inicio de sesión y obtención del token JWT.
- Crear el método para enviar la solicitud a través del API Gateway con el token JWT en el encabezado.
- Crear el programa principal, iniciar sesión y obtener los productos a través del API Gateway

Con esta práctica, se ha implementado un servicio de autenticación que emite tokens JWT con más control y funcionalidades adicionales, como el inicio de sesión y la verificación del token. También se ha protegido los microservicios utilizando el API Gateway (Kong) y asegurando que solo los usuarios autenticados pueden obtener información.





Práctica 2. Aplicar seguridad a microservicios implementando mecanismos de autenticación. Utilizar Python con Flask para los microservicios y MongoDB como base de datos para la gestión de usuarios.

1. Configuración de MongoDB

- Crear una cuenta para acceder a MongoDB a través de Atlas, crear la base de datos y la colección usuarios, películas.

2. Crear el servicio de "Usuarios" para la autenticación

- Importar las librerías: Flask, jwt, datetime, MongoClient, ObjectId.
- Crear la clave secreta para firmar y verificar el token JWT.
- Crear la conexión a la base de datos.
- Crear la ruta para hacer login y su método para Generar el token JWT con el Id de usuario como payload.
- Establecer el puerto HTTP como el 5001.

3. Crear el servicio de "Películas" protegido con autenticación y autorización mediante JWT.

- Importar las librerías: Flask, jwt, datetime, MongoClient, ObjectId.
- Crear la clave secreta para firmar y verificar el token JWT.
- Crear la conexión a la base de datos.
- Crear el método para verificar el token JWT.
- Crear la ruta para obtener las películas, aplicar la autorización y autenticación.
- Establecer el puerto HTTP como el 5000.

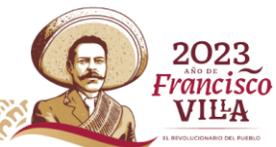
4. Realizar las peticiones para registrar usuarios y películas

- Crear algunos registros de usuarios en la colección "usuarios" y registros de películas en la colección "películas" para poder realizar las pruebas de autenticación y acceso a las películas.

5. Realizar las peticiones a través del servicio de "Usuarios" para obtener el token JWT.

6. Utilizar el token JWT para acceder al servicio de "Películas".

- Realizar la petición para obtener las películas utilizando el token JWT





Práctica 3. Implementar la gestión de identidades y usuarios utilizando Keycloak con tecnología Facebook o Google como proveedores de identidades en una aplicación de ejemplo con Python y Flask.

1. Configuración de Keycloak

- Descargar la imagen de Docker de Keycloak desde el repositorio oficial: Docker pull jboss/keycloak
- Crear un archivo 'docker-compose.yml' para configurar el contenedor de keycloak

2. Configurar el realm en Keycloak con proveedores de identidades externos

- Modificar el archivo 'realm-export.json' para agregar configuraciones de proveedores de identidades externos (Facebook o Google).
- Reemplazar el 'id_de_facebook' y el 'secret_key' con las credenciales proporcionadas por Facebook para la aplicación.

3. Iniciar Keycloak en Docker

- Ejecutar 'docker-compose up -d' para iniciar Keycloak en el contenedor de Docker.

4. Crear el servicio de "Usuarios" con Flask

- Importar las librerías: Flask, jwt, datetime, etc.
- Crear la ruta y el método para implementar la lógica para autenticar usuarios con Facebook utilizando sdk para su autenticación, generar el token JWT y regresarlo.
- Establecer el puerto HTTP en 5001.

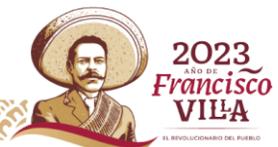
5. Crear el servicio de "Películas" protegido con autenticación mediante Keycloak

- Importar las librerías: Flask, jwt, etc.
- Crear el método para verificar el token.
- Crear la ruta y el método para obtener las películas, es necesario verificar el usuario.
- Establecer el puerto HTTP en 5000.

Tema V

Práctica 1. Aplicar Docker y microservicios para crear una pequeña aplicación web de lista de tareas. Para simplificar, se implementarán dos microservicios: uno para gestionar las tareas y otro para el frontend de la aplicación.

1. Configurar la estructura de archivos





- Instalar Docker en el sistema
 - Crear una nueva carpeta llamada 'microservicios_todo_app' y dentro de ella crear dos subcarpetas: 'backend' y 'frontend'.
2. Backend - Microservicio para Gestionar Tareas
- Dentro de la carpeta 'backend', crear un archivo llamado 'app.py'.
 - Importar dentro del archivo 'app.py' las librerías: Flask, jsonify, request, etc.
 - Crear la ruta y el método para agregar una tarea.
 - Establecer el puerto HTTP en 5000.
3. Microservicio para el Frontend
- Dentro de la carpeta 'frontend', crea un archivo llamado 'index.html'.
 - Crear la estructura del documento HTML para que cuente con un cuadro de texto y un botón de comando y una lista para desplegar la lista de tareas.
 - Crear el script en javascript, utilice una función asíncrona para que realice una petición al backend realizado en el punto anterior e imprima en el front la lista de tareas que regresa el back.
4. Creación de Dockerfiles
- Dentro de las carpetas 'backend' y 'frontend', crea un archivo llamado 'Dockerfile'.
5. Archivo requirements.txt
- Dentro de la carpeta 'backend', crear el archivo 'requirements.txt'.
 - Agregue las librerías o paquetes necesarios: Flask==2.0.1.
- 6: Construcción y Ejecución de los Contenedores
- Abrir una terminal y asegurarse de estar en la carpeta raíz, ejecutar los comandos para construir y ejecutar los contenedores.
 - Teclear los comandos para construir y ejecutar el contenedor del backend, frontend.
7. Acceso a la Aplicación
- Abrir el navegador web y visita 'http://localhost'. Puede agregar nuevas tareas en el campo de texto y haciendo clic en el botón "Add Task". Las tareas se almacenarán en el backend en el microservicio correspondiente. Con esta práctica, ha implementado dos microservicios usando Docker: uno para el backend y otro para el frontend. La aplicación de lista de tareas ahora se ejecuta en contenedores independientes y puede acceder a ella desde tu navegador web.





Práctica 2. Escalabilidad Automatizada con Kubernetes. En esta práctica, se utilizará Kubernetes como orquestador de contenedores para lograr la escalabilidad automatizada de nuestros microservicios.

1. Configuración de Kubernetes

- Instalar y configurar Kubernetes en el entorno de nube o local. Usar soluciones como Minikube para una configuración local o servicios administrados de Kubernetes ofrecidos por proveedores de la nube como Google Kubernetes Engine (GKE) o Amazon Elastic Kubernetes Service (EKS).

2. Despliegue de Microservicios

- Crear y empaquetar los microservicios en contenedores Docker siguiendo las prácticas de diseño mencionadas anteriormente.
- Define los archivos de configuración de Kubernetes (deployment.yaml) para cada microservicio. Especifica el número deseado de réplicas y los recursos (CPU, memoria) asignados a cada instancia.

3. Configurar el Autoescalado

- Configurar las políticas de autoescalado en el archivo deployment.yaml para escalar automáticamente el número de réplicas de un microservicio en función de la carga de trabajo.

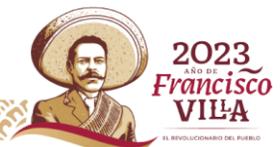
4. Despliegue y Monitoreo

- Implementar los microservicios en Kubernetes.
- Monitorear el comportamiento y la salud de tus microservicios utilizando herramientas como Kubernetes Dashboard, Prometheus y Grafana. Establecer alertas para detectar problemas y tomar medidas proactivas.

Práctica 3. Implementación de un Microservicio en la Nube con Google Cloud Platform y Flask.

1. Configuración de Google Cloud Platform

- Crear una cuenta en Google Cloud Platform (GCP) y crea un nuevo proyecto.
- Instalar el SDK de Google Cloud en tu máquina local para interactuar con GCP desde la línea de comandos.
- Autentica el SDK con tu cuenta de GCP usando el comando 'gcloud auth login'.





2. Configurar el Proyecto de Flask

- Crear una nueva carpeta en la máquina local y navega a ella usando la línea de comandos.
- Instalar Flask en tu entorno virtual de Python.

3. Microservicio saludar

- Crear un archivo llamado 'app.py' en la carpeta que creaste anteriormente.
- Abre el archivo 'app.py' con tu editor de código preferido, :
- Importar la librerías: Flask, request, etc.
- Crear la ruta y el método que retorne un mensaje de bienvenida.
- Establecer el puerto HTTP a 8080.

4. Desplegar el microservicio en GCP

- Crear un archivo llamado 'Dockerfile' en la misma carpeta que 'app.py' y escribir el escript para copiar y ejecutar los requerimientos.
- Crear el archivo llamado 'requirements.txt' en la misma carpeta, el contenido serán los requerimientos de la aplicación en este caso flask==2.0.1
- Construir la imagen de Docker del microservicio.
- Etiquetar la imagen y subirla a Google Container Registry (GCR).

5. Desplegar el Microservicio en GKE

- Acceder a la Consola de GCP y abrir la página de Kubernetes Engine.
- Crear un nuevo clúster de GKE y seleccionar las opciones de configuración que mejor se adapten a tus necesidades.
- Descargar las credenciales para conectarse a él.
- Desplegar el microservicio en el clúster.
- Exponer el microservicio como un servicio de Kubernetes.

6. Acceso al Microservicio

- Esperar unos momentos para que el balanceador de carga se configure.
- Acceder al microservicio en tu navegador web o usando una herramienta como CURL o Postman.





9. Proyecto de asignatura

Fundamento:

El objetivo del proyecto es desarrollar en equipo una propuesta de una aplicación full-stack que dé solución a un problema real o proceso que requiera la aplicación de un programa móvil, web y/o de escritorio. Los proyectos a realizar se establecen a partir de especificaciones que realice el docente o necesidades que los estudiantes determinen que existen en el entorno.

La aplicación debe implementar microservicios para gestionar los usuarios, productos, notificaciones, etc. Los microservicios dependerán del tipo de aplicación que se desee generar, implementar la seguridad de la aplicación con autenticación u autorización en tokens JWT o OAuth 2.0 y desplegar los microservicios en la nube para manejar cargas de trabajo viables, asegurar su escalabilidad y disponibilidad y garantizar su disponibilidad.

Planeación:

Los equipos serán de tres integrantes como máximo, el proyecto a realizar podrá considerar en su desarrollo las competencias adquiridas en asignaturas relacionadas con el desarrollo de software, así como los principios del desarrollo ágil. Sin olvidar tanto elementos de seguridad como de verificación de la calidad del proyecto.

Desarrollo:

Durante el desarrollo se llevarán a cabo las actividades planteadas en la planeación y a las necesidades del proyecto, pero se desarrollan de forma particular algunas propuestas que deben de cumplirse.

La aplicación debe contemplar:

1. Análisis y Planificación:

- Definición de requerimientos: Identifica los objetivos del proyecto, las funcionalidades necesarias y los usuarios a los que va dirigido.
- Planificación del proyecto: Estima el tiempo y recursos necesarios, establece fechas clave y define las tareas a realizar.

2. Diseño:

- Diseño de la arquitectura: Decide cómo se organizarán los microservicios y cómo se comunicarán entre sí.





- Diseño de la base de datos: Define la estructura de la base de datos y cómo se relacionarán las entidades.

- Diseño de la interfaz de usuario: Crea los wireframes o prototipos para definir la apariencia y funcionalidades de la interfaz.

3. Implementación:

- Desarrollo de microservicios: Crea los microservicios utilizando el lenguaje de programación y el framework adecuado. Implementa las funcionalidades específicas de cada microservicio.

- Implementación del frontend: Desarrolla el frontend de la aplicación utilizando HTML, CSS y JavaScript, y comunícalo con los microservicios a través de APIs.

4. Pruebas:

- Pruebas unitarias: Realiza pruebas en cada microservicio para asegurar que funcionen correctamente de manera individual.

- Pruebas de integración: Verifica que los microservicios se comuniquen correctamente entre sí y que la integración con el frontend sea exitosa.

- Pruebas de rendimiento: Evalúa el rendimiento de la aplicación y su capacidad de manejar diferentes cargas de tráfico.

5. Seguridad:

- Implementación de seguridad: Asegura la autenticación y autorización en los microservicios. Utiliza HTTPS para la comunicación segura.

- Pruebas de seguridad: Realiza pruebas de seguridad para identificar vulnerabilidades y mitigar posibles riesgos.

6. Despliegue:

- Despliegue en entornos de prueba y producción: Despliega los microservicios y el frontend en un entorno de prueba para verificar que todo funcione correctamente antes de pasar a producción.

- Despliegue en la nube: Utiliza plataformas de la nube como Google Cloud Platform, Microsoft Azure o Amazon Web Services para implementar tus microservicios.

7. Monitoreo y Mantenimiento:

- Configuración de monitoreo: Establece herramientas de monitoreo para rastrear el rendimiento y la disponibilidad de los microservicios.





- Actualizaciones y mantenimiento: Realiza actualizaciones y mejoras según sea necesario para mantener la aplicación en buen estado.

8. Documentación:

- Documentación técnica: Crea documentación clara y detallada de la arquitectura, el diseño y la implementación de los microservicios.

- Documentación de usuario: Proporciona guías y manuales de usuario para facilitar la adopción de la aplicación.

9. Lanzamiento y Evaluación:

- Lanzamiento: Pasa la aplicación a producción y asegúrate de que esté disponible para los usuarios.

- Evaluación continua: Realiza evaluaciones periódicas para identificar áreas de mejora y realizar ajustes según los comentarios de los usuarios.

Evaluación

La rúbrica contemplará los siguientes criterios:

Planeación 20%

Desarrollo 40%

Publicación 10%

Exposición del proyecto 20%

Trabajo en equipo 10%

10. Evaluación por competencias

La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño encada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Rúbricas o productos, señalados en cada unidad académica dentro de las actividades de aprendizaje.
- Prácticas propuestas y su presentación y exposición en plenaria. Algunas se evaluarán por equipo.
- Portafolio de evidencias de Información recabada durante las consultas e investigaciones solicitadas, plasmadas en documentos escritos.

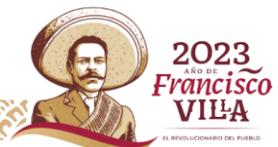




- Descripción de otras experiencias concretas que se obtendrán al participar en discusiones, exposiciones o cualquier otro medio didáctico-profesional que trate sobre la materia y que deberán realizarse durante el curso académico.
- Exámenes teórico-prácticos para comprobar la efectividad del estudiante en la resolución de casos prácticos.

11. Fuentes de información

1. Chris Richardson, (2020) Microservices Patterns: With examples in Java 1ra Edición.
2. Boris Scholl, Jonas Bonér, y Eberhard Wolff, (Junio 2016) Implementing Microservices: The Practical Guide to Building Enterprise Applications.
3. Tarek Ziade y Samuele Pedroni, (Octubre 2019) Python Microservices Development, 1ra Edición.
4. Nikhil Ketkar, (Marzo 2020) Python Microservices with Flask: Build, test, and deploy scalable services with Flask, Docker, and Kubernetes 1ra Edición.
5. Tarek Ziade y Salem Mohamed, (Diciembre 2017) Python Microservices Development Cookbook: Over 80 recipes for building scalable microservices using Python and Flask 1ra Edición.
6. Tarek Ziade, (Agosto 2019) Python Microservices Web Development: With Django, Third Edition 3ra Edición.
7. Gene Kim, Patrick Debois, John Willis, y Jez Humble, (Octubre 2016) The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations 1ra Edición.





1. Datos Generales de la asignatura

Nombre de la asignatura:	Visión por Computadora y Machine Learning.
Clave de la asignatura:	DAM-2403
SATCA¹:	2-4-6
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

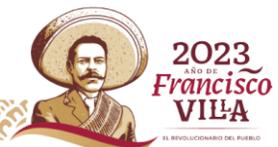
Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las competencias profesionales:

- Aplicar conocimientos matemáticos y estadísticos en la solución de problemas de visión por computadora y machine learning con un enfoque interdisciplinario.
- Implementar el procesamiento de imágenes a través de manipulación, segmentación, filtrado, transformaciones y extracción de características.
- Analizar, modelar, desarrollar, implementar sistemas de visión por computadora por lo que debe tener una comprensión de los conceptos y técnicas de visión por computadora, como detección de objetos, reconocimiento facial, seguimiento de objetos y segmentación semántica.
- Abordar problemas complejos y tener habilidades analíticas para analizar, diseñar e implementar soluciones efectivas en el campo de visión por computadora y machine learning.

La importancia de esta asignatura, es que permite al estudiante crear e implementar modelos de inteligencia artificial en la nube capaces de resolver problemas complejos en diversos sectores, con capacidad para analizar datos visuales, potenciales para la automatización y eficiencia, para después consumir estos modelos a través de servicios en aplicaciones con tecnologías en las que hoy nos conectamos desde teléfonos móviles, tabletas, eBooks, netbooks, computadoras y otra gama de dispositivos.

¹ Sistema de Asignación y Transferencia de Créditos Académicos





Intención didáctica

La asignatura debe ser práctica y capaz de desarrollar en el estudiante la habilidad para desarrollar habilidades en el análisis y procesamiento de imágenes, aplicar técnicas de reconocimiento de patrones y aprendizaje automático, diseño y entrenamiento de redes neuronales convolucionales. Así también fomentar la creación de modelos y su despliegue en la nube con el lenguaje Python y frameworks y librerías de Inteligencia Artificial.

- En el tema I. El estudiante será capaz de adquirir conocimiento y técnicas para cargar, limpiar, transformar y procesar datos; así como explorar, identificar patrones, tendencias y relaciones entre variables y obtener información y conocimientos a partir de éstos. utilizando herramientas disponibles en Python, implica el uso de librerías como Pandas, NumPy y Matplotlib para realizar visualizaciones, cálculos estadísticos y generar gráficos descriptivos.
- En el tema II. El estudiante será capaz de adquirir conocimientos sólidos sobre los principios y conceptos clave del Deep Learning incluyendo las redes neuronales artificiales, las capas, las funciones de activación y el proceso de aprendizaje. Aplicar el Deep Learning en problemas de clasificación y reconocimiento, como la clasificación de imágenes, el reconocimiento de voz, el procesamiento de texto o la detección de objetos. Desarrollar una perspectiva crítica sobre el Deep Learning, comprendiendo sus limitaciones, desafíos y posibles sesgos. Esto implica aprender a evaluar la calidad y robustez de los modelos, a interpretar sus resultados y a tomar decisiones informadas sobre su uso en diferentes contextos y aplicaciones.
- En el tema III. El estudiante adquirirá conocimientos sólidos sobre la arquitectura y el funcionamiento de las redes neuronales convolucionales, incluyendo conceptos como las capas convolucionales, las capas de agrupación (pooling), las funciones de activación y las capas completamente conectadas. También, adquirirá experiencia práctica en la implementación de redes neuronales convolucionales utilizando frameworks y bibliotecas populares como TensorFlow y Keras. Esto implica aprender a diseñar, entrenar y evaluar modelos de redes neuronales convolucionales, así como a ajustar sus hiperparámetros y utilizar técnicas de regularización y optimización. Desarrollar una perspectiva crítica sobre las redes neuronales convolucionales, comprendiendo sus limitaciones, desafíos y posibles sesgos.
- En el tema IV. Adquirirá conocimientos sobre la arquitectura y el funcionamiento de las redes neuronales recurrentes, incluyendo conceptos como las células de memoria





recurrente (por ejemplo, las LSTM o GRU), el mecanismo de retroalimentación temporal y la capacidad de procesar secuencias de datos. Utilizar los conocimientos adquiridos para aplicar redes neuronales recurrentes en problemas de procesamiento del lenguaje natural, como la clasificación de texto, la generación de texto, la traducción automática o el modelado de lenguaje. Esto implica aprender a procesar y representar secuencias de texto, seleccionar arquitecturas de redes neuronales recurrentes adecuadas y utilizar técnicas como la atención y la transferencia de aprendizaje.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Jiquilpan Jiquilpan Michoacán, 08 de mayo de 2023	Lic. Martha Elizabeth González Zepeda M. C. Edna Paola Chávez Guerra M. C. María Elena Cabrera Rocha Ing. Leonardo Martínez González	Diseño y desarrollo de los programas de estudio del módulo de especialidad para la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<ul style="list-style-type: none"> Aplicar eficazmente técnicas y algoritmos de visión por computadora y machine learning para analizar, interpretar y extraer información significativa de imágenes y datos visuales, con el fin de resolver problemas complejos en diversos dominios, como reconocimiento de objetos, detección de patrones, segmentación, clasificación, seguimiento y generación automática de contenido visual Además, implica la capacidad de abordar problemas complejos, adaptar y combinar diferentes técnicas y algoritmos, seleccionar y preprocesar adecuadamente los datos, evaluar y optimizar los modelos desarrollados, y comunicar los resultados obtenidos de manera clara y efectiva.





5. Competencias previas

Esta asignatura, es la aplicación práctica del conocimiento matemático y estadístico, de programación orientada a objetos, lógica y funcional, de fundamentos del machine learning y procesamiento de imágenes a través de los métodos y técnicas adecuados para la implementación de modelos de inteligencia artificial y su despliegue en la nube para después consumirlo como un servicio.

Esta asignatura tiene relación previa con las signaturas:

- Programación orientada a objetos.
- Inteligencia artificial.
- Programación web.
- Arquitectura orientada a servicios

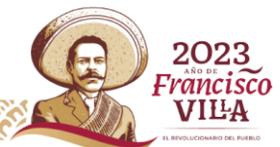
Considerando las asignaturas anteriormente mencionadas se consideran las siguientes competencias específicas:

- Aplicar un lenguaje orientado a objetos para la solución de problemas.
- Conocer los principios y el desarrollo de la Inteligencia Artificial, identificando sus aplicaciones (robótica, visión computacional, lógica difusa, redes neuronales y procesamiento de lenguaje natural) para emplearlas en el diseño e implementación de sistemas inteligentes que faciliten las tareas del ser humano. Modelar casos de uso acorde a los requerimientos del proyecto.
- Desarrollar aplicaciones web que involucre lenguajes de marcas, de presentación, del lado del cliente, del lado del servidor, con la integración de servicios web.
- Documentar el proyecto.

Tomar decisiones con base en los elementos teórico-práctico adquiridos que permitan optimizar costos en soluciones informáticas bajo ambiente Web.

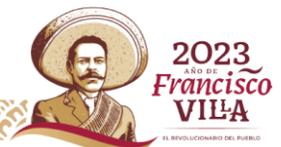
6. Temario

No.	Temas	Subtemas
-----	-------	----------





1	Procesamiento y Análisis de Datos.	<ul style="list-style-type: none"> 1.1. Matrices. <ul style="list-style-type: none"> 1.1.1. Arreglos unidimensionales y multidimensionales. 1.1.2. Creación de arreglos. 1.1.3. Operaciones aritméticas con arreglos. 1.1.4. Uso de índices en arreglos. 1.1.5. Programación orientada a los arreglos. 1.1.6. Funciones universales. 1.2. Graficación de información. <ul style="list-style-type: none"> 1.2.1. Figuras y subplots. 1.2.2. Gestión de gráficas. 1.2.3. Guardado de gráficas en archivos. 1.3. Manipulación y tratamiento de datos. <ul style="list-style-type: none"> 1.3.1. Series y matrices en dos dimensiones. 1.3.2. Indizado, selección y filtrado. 1.3.3. Aritmética y alineamiento de datos. 1.3.4. Manipulación algebraica de datos.
2	Fundamentos de Deep Learning.	<ul style="list-style-type: none"> 2.1. Qué es Deep Learning. <ul style="list-style-type: none"> 2.1.1. ¿Por qué Deep Learning? 2.1.2. ¿Por qué ahora? 2.2. Redes neuronales. <ul style="list-style-type: none"> 2.2.1. Anatomía de la red neuronal. 2.2.2. Componentes matemáticos de las redes neuronales. 2.2.3. Librerías para gestión de redes neuronales. 2.3. Perceptrones multicapa - MLP. 2.4. Aprendizaje supervisado. 2.5. Aprendizajes no supervisados. 2.6. Algoritmos de aprendizajes <ul style="list-style-type: none"> 2.6.1. Descenso por gradiente.





		<p>2.6.2. Backpropagation. 2.6.3. Hiperparámetros. 2.6.4. Métricas de desempeño. 2.7. Creación de modelos de redes neuronales. 2.8. Funciones de activación para redes neuronales multicapa. 2.9. Despliegue de modelos.</p>
3	Redes Neuronales Convolucionales.	<p>3.1. ¿Qué es la convolución? 3.2. Convolución 2D. 3.3. Convolución 3D. 3.4. Filtros de procesamiento de imágenes. 3.5. Componentes de una red neuronal convolucional. 3.6. Retropropagación a través de la capa convolucional. 3.7. Retropropagación a través de las capas de agrupación. 3.8. Técnicas de regulación. 3.8.1. Dropout. 3.8.2. Data augmentation. 3.9. Arquitectura de redes neuronales convolucionales. 3.10. Transferencia de conocimientos. 3.11. Creación de modelos con redes neuronales convolucionales.</p>
4	Redes Neuronales Recurrentes.	<p>4.1. Redes generativas antagónicas GAN. 4.1.1. Autoencoders como predecesores. 4.1.2. Mecanismos de las redes generativas y discriminativas. 4.2. Redes neuronales recurrentes RNN. 4.2.1. El mecanismo de las redes recurrentes.</p>





		<p>4.2.2. Tipos de RNN: LSTM y GRU.</p> <p>4.2.3. Procesamiento del lenguaje natural – NLP.</p> <p>4.2.4. Bolsa de palabras y Word embedding.</p> <p>4.3. Creación de modelos para aplicaciones de lenguaje natural.</p>
--	--	--

7. Actividades de aprendizaje de los temas

I Procesamiento y Análisis de Datos.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Proporcionar habilidades prácticas en programación, resolución de problemas, manipulación de datos numéricos, análisis de datos y optimización del rendimiento. • Adquirir habilidades esenciales en visualización de datos, análisis exploratorio, comunicación efectiva y presentación de resultados. • Tener habilidades en manipulación de datos, preparación de datos y trabajo con datos estructurados. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. 	<ul style="list-style-type: none"> • Realizar un cuadro de las librerías utilizadas para manipulación de datos numéricos, visualización y preparación de datos estructurados. • Instalar un entorno local interactivo de Python para el análisis, visualización y manipulación de datos estructurados. • Instalación y configuración de un ecosistema de Python en la nube. • Realizar prácticas de carga de imágenes, manipulaciones y transformaciones de datos utilizando NumPy y Pandas, y visualizar resultados utilizando Matplotlib. • Realizar prácticas de limpieza de datos, de normalización, extracción de características y manejo de datos faltantes utilizando Pandas y NumPy. • Implementar algoritmos de clasificación, regresión u otros en Python utilizando estas bibliotecas.



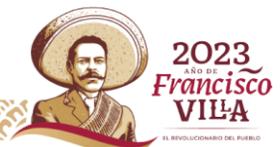


<ul style="list-style-type: none"> Comunicación oral y escrita. 	
II Fundamentos de Deep Learning.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Proporcionar una base sólida para trabajar en proyectos de aprendizaje profundo y aprovechar las oportunidades en campos como la inteligencia artificial, la visión por computadora y el procesamiento de lenguaje natural. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Capacidad de aplicar los conocimientos en la práctica. Toma de decisiones. 	<ul style="list-style-type: none"> Investigar los conceptos fundamentales del Deep Learning: redes neuronales, capas, funciones de activación, propagación hacia atrás (backpropagation), optimización y regularización. Realizar en un cuadro sinóptico la clasificación de las tablas en HTML. Implementar el framework Tensorflow, Pytorch o Keras, experimenta con modelos de aprendizaje profundo de manera práctica. Resolver un problema de clasificación de imágenes y trabaja en la implementación de modelos de aprendizaje profundo. Desplegar el modelo en la nube y consumirlo desde una aplicación web y/o, móvil. Investigar y leer papers académicos para comprender los últimos avances y técnicas en el campo del Deep Learning.
III Redes Neuronales Convolucionales.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Desarrollar habilidades y conocimientos específicos en el 	<ul style="list-style-type: none"> Investigar los conceptos básicos, como las capas convolucionales, de agrupación y completamente conectadas, así como las





<p>diseño, implementación y aplicación de redes neuronales convolucionales en el campo del aprendizaje automático y la visión por computadora.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. • Capacidad de aplicar los conocimientos en la práctica. 	<p>funciones de activación utilizadas en las CNN.</p> <ul style="list-style-type: none"> • Aplicar los conocimientos de redes neuronales convolucionales en proyectos prácticos de visión por computadora. Experimentar con diferentes arquitecturas y técnicas de entrenamiento para mejorar la precisión y el rendimiento del modelo. • Explorar arquitecturas CNN más avanzadas, como ResNet, Inception, VGG o DenseNet. Estudiar su estructura y los principios en los que se basan. • Aplicar transferencia de conocimiento. Utilizar redes preentrenadas en conjuntos de datos específicos. Cargar modelos CNN preentrenados, como los entrenados en ImageNet, y ajústalos a problemas de clasificación o detección de objetos en conjuntos propios de datos. • Participar en competencias o desafíos relacionados con la clasificación de imágenes o la detección de objetos utilizando redes CNN.
IV Redes Neuronales Recurrentes.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Desarrollar habilidades sólidas en el uso de redes neuronales recurrentes para el procesamiento del lenguaje natural, la traducción automática, la generación de texto. 	<ul style="list-style-type: none"> • Investigar los conceptos fundamentales: unidades LSTM y GRU, así como las operaciones de actualización y olvido en una RNN. Principios de propagación hacia adelante y hacia atrás en las redes recurrentes. • Implementar redes neuronales recurrentes básicas utilizando frameworks populares

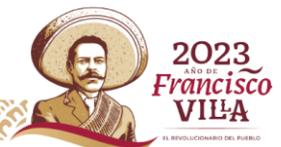




<p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. • Capacidad de aplicar los conocimientos en la práctica. 	<p>como TensorFlow o PyTorch. Comenzar con ejemplos simples de secuencias y textos para entender cómo se propagan las señales a través del tiempo y cómo se actualizan los estados ocultos en una RNN.</p> <ul style="list-style-type: none"> • Experimentar con diferentes arquitecturas. Explorar diferentes arquitecturas de redes neuronales recurrentes más avanzadas, como las RNN bidireccionales o las RNN apiladas. • Aplicar RNN en tareas de procesamiento de lenguaje natural. Utiliza RNN para resolver problemas de procesamiento del lenguaje natural, como la generación de texto o la traducción automática. • Participar en proyectos prácticos y competencias relacionadas con el procesamiento de lenguaje natural y las redes neuronales recurrentes.
---	---

8. Práctica(s)

<p>Tema I</p> <p>Práctica 1. Análisis exploratorio de datos de ventas mensuales.</p> <ol style="list-style-type: none"> 1. Utiliza Pandas para cargar un conjunto de datos que contenga información mensual de ventas. 2. Utiliza Pandas y NumPy para limpiar y preparar los datos, eliminando filas o columnas innecesarias y tratando los valores faltantes si los hay. 3. Utiliza NumPy para calcular estadísticas descriptivas de las ventas, como el total de ventas, promedio mensual, máximo y mínimo. 4. Utiliza Matplotlib para crear gráficos como un gráfico de líneas que muestre la tendencia de las ventas a lo largo del tiempo y un gráfico de barras que muestre las ventas mensuales. 5. Agrega etiquetas, títulos y leyendas a los gráficos para mejorar la presentación y la comprensión de los datos.





Práctica 2. Preprocesamiento de datos de imágenes para clasificación.

1. Utiliza Pandas para cargar un conjunto de datos que contenga imágenes y sus respectivas etiquetas de clasificación.
2. Utiliza NumPy para cargar y procesar las imágenes, convirtiéndolas en matrices numéricas y realizando operaciones de normalización si es necesario.
3. Utiliza Pandas para realizar transformaciones en las etiquetas, como codificarlas en valores numéricos si son categóricas.
4. Utiliza Matplotlib para visualizar algunas de las imágenes antes y después del preprocesamiento para asegurarte de que los datos estén correctamente cargados y listos para su uso.
5. Utiliza las funciones de división de datos de Pandas y NumPy para dividir los datos en conjuntos de entrenamiento y prueba antes de aplicar modelos de clasificación.

Práctica 3. Análisis de series de tiempo de datos climáticos.

1. Utiliza Pandas para cargar un conjunto de datos que contenga registros climáticos a lo largo del tiempo.
2. Utiliza Pandas para realizar cualquier manipulación necesaria en los datos, como cambiar el formato de fechas o filtrar datos irrelevantes.
3. Utiliza NumPy para realizar cálculos en los datos, como el cálculo de promedios móviles o diferenciaciones.
4. Utiliza Matplotlib para crear gráficos que muestren patrones en los datos climáticos, como gráficos de líneas para la evolución de la temperatura a lo largo del tiempo.
5. Agrega elementos visuales a los gráficos, como sombreado para resaltar intervalos de tiempo o líneas de referencia para mostrar promedios históricos.

Práctica 4. Detección y reconocimiento de placas en tiempo real.

1. Instalación de bibliotecas
Paso 1: Asegúrate de tener Python instalado en tu sistema.
Paso 2: Instala las bibliotecas necesarias, como OpenCV, NumPy y Tesseract OCR. Puedes usar pip para instalarlos.
2. Preparación del modelo y los archivos de configuración
Paso 1: Descarga el modelo pre-entrenado para la detección de placas de vehículos, como el modelo YOLOv3, y los archivos de configuración correspondientes.
Paso 2: Configura los archivos de configuración para establecer los parámetros adecuados, como el umbral de confianza y los nombres de las clases.





3. Configuración de la cámara

Paso 1: Conecta la cámara al sistema o utiliza una fuente de video en tiempo real.

Paso 2: Configura los parámetros de la cámara, como la resolución y la velocidad de cuadros, si es necesario.

4. Detección de placas en tiempo real

Paso 1: Carga el modelo pre-entrenado y los archivos de configuración en tu aplicación de Python utilizando OpenCV.

Paso 2: Inicializa el objeto de detección de objetos y configura los parámetros necesarios.

Paso 3: Captura el video en tiempo real y, en cada cuadro, realiza la detección de objetos utilizando el modelo cargado.

Paso 4: Filtra las detecciones para obtener solo las detecciones de placas de vehículos.

Paso 5: Dibuja un cuadro delimitador alrededor de cada placa detectada en el cuadro de video.

5. Reconocimiento de texto en las placas detectadas

Paso 1: Recorta y preprocesa cada región de la placa detectada para obtener una imagen de la placa individual.

Paso 2: Aplica OCR (reconocimiento óptico de caracteres) a la imagen de la placa utilizando Tesseract OCR u otra biblioteca de tu elección.

Paso 3: Obtén el texto reconocido y realiza cualquier postprocesamiento necesario, como limpieza de ruido y formato de salida.

6. Mostrar resultados en tiempo real

Paso 1: Muestra el video en tiempo real con los cuadros delimitadores y los textos reconocidos superpuestos.

Paso 2: Actualiza continuamente la visualización en tiempo real para mostrar los resultados de la detección y el reconocimiento.

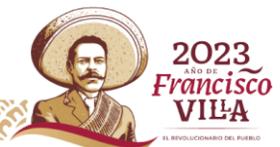
Tema II

Practica 1 Reconocimiento facial, utilizando redes neuronales

1. Preparación de datos

Paso 1: Importa las bibliotecas necesarias, como TensorFlow o PyTorch, y carga el conjunto de datos de imágenes faciales etiquetadas.

Paso 2: Preprocesa los datos de imágenes, como el cambio de tamaño, la normalización de los valores de píxeles y la codificación de etiquetas.





Paso 3: Divide el conjunto de datos en conjuntos de entrenamiento y prueba.

2. Construcción de la red neuronal

Paso 1: Diseña y construye una red neuronal para el reconocimiento facial, definiendo la arquitectura de la red neuronal con capas densas y funciones de activación.

Paso 2: Asegúrate de incluir una capa de salida con un número de neuronas igual al número de clases de reconocimiento facial.

3. Entrenamiento del modelo

Paso 1: Compila el modelo con la función de pérdida adecuada y el optimizador, y define las métricas de evaluación.

Paso 2: Entrena el modelo utilizando el conjunto de datos de entrenamiento y ajusta los pesos de la red neuronal mediante el proceso de retropropagación.

4. Evaluación del modelo

Paso 1: Evalúa el modelo utilizando el conjunto de datos de prueba y calcula métricas de rendimiento, como la precisión y la matriz de confusión.

Paso 2: Realiza predicciones con el modelo entrenado en nuevas imágenes faciales y verifica su precisión.

5. Detección y reconocimiento facial en tiempo real

Paso 1: Utiliza bibliotecas y técnicas de visión por computadora, como OpenCV, para capturar video en tiempo real desde una cámara.

Paso 2: Procesa los cuadros de video para detectar rostros utilizando técnicas de detección facial, como el detector de Haar o el detector de puntos de referencia faciales.

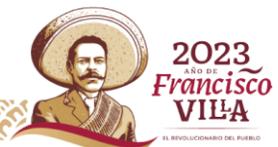
Paso 3: Extrae las regiones faciales detectadas y aplica preprocesamiento, como el cambio de tamaño y la normalización de los valores de píxeles.

Paso 4: Utiliza el modelo entrenado para realizar el reconocimiento facial en las regiones faciales y muestra el resultado, como el nombre o la etiqueta asociada a cada rostro reconocido.

Practica 2 Reconocer el sexo de una persona.

1. Preparación de datos

Paso 1: Importa las bibliotecas necesarias, como TensorFlow o PyTorch, y carga el conjunto de datos de imágenes etiquetadas con información de sexo.





Paso 2: Preprocesa los datos de imágenes, como el cambio de tamaño y la normalización de los valores de píxeles.

Paso 3: Codifica las etiquetas de sexo, asignando un valor numérico (por ejemplo, 0 para masculino y 1 para femenino).

Paso 4: Divide el conjunto de datos en conjuntos de entrenamiento y prueba.

2. Construcción de la red neuronal

Paso 1: Diseña y construye una red neuronal para el reconocimiento del sexo, definiendo la arquitectura de la red neuronal con capas densas y funciones de activación.

Paso 2: Asegúrate de incluir una capa de salida con una neurona para clasificar entre los sexos.

3. Entrenamiento del modelo

Paso 1: Compila el modelo con la función de pérdida adecuada y el optimizador, y define las métricas de evaluación.

Paso 2: Entrena el modelo utilizando el conjunto de datos de entrenamiento y ajusta los pesos de la red neuronal mediante el proceso de retropropagación.

4. Evaluación del modelo

Paso 1: Evalúa el modelo utilizando el conjunto de datos de prueba y calcula métricas de rendimiento, como la precisión y la matriz de confusión.

Paso 2: Analiza los resultados para determinar la efectividad del modelo en el reconocimiento del sexo de una persona.

5. Reconocimiento del sexo en imágenes

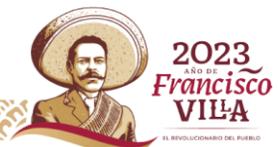
Paso 1: Utiliza bibliotecas y técnicas de procesamiento de imágenes, como OpenCV, para cargar y preprocesar imágenes de personas.

Paso 2: Utiliza el modelo entrenado para predecir el sexo de una persona en una imagen.

Paso 3: Muestra el resultado del reconocimiento, indicando el sexo predicho para la persona en la imagen.

Practica 3 Análisis de sentimientos en comentarios y clasificarlos como positivos o negativos

1. Preparación de datos





Paso 1: Importa las bibliotecas necesarias, como TensorFlow o PyTorch, y carga el conjunto de datos etiquetado con comentarios y sus respectivas etiquetas de sentimiento (positivo o negativo).

Paso 2: Realiza una limpieza de los datos, eliminando caracteres especiales, números o cualquier información no relevante para el análisis de sentimientos.

Paso 3: Dividir el conjunto de datos en conjuntos de entrenamiento y prueba.

2. Procesamiento de texto

Paso 1: Realiza la tokenización de los comentarios, dividiendo el texto en palabras individuales o unidades de texto más pequeñas.

Paso 2: Crea un vocabulario, asignando un índice numérico a cada palabra única presente en los comentarios.

Paso 3: Convierte los comentarios en secuencias de números utilizando el vocabulario creado.

3. Construcción de la red neuronal

Paso 1: Diseña y construye una red neuronal para el análisis de sentimientos, utilizando capas de embeddings para representar las palabras y capas densas para la clasificación.

Paso 2: Define la arquitectura de la red neuronal, incluyendo la cantidad de capas ocultas, las funciones de activación y la capa de salida.

4. Entrenamiento del modelo

Paso 1: Compila el modelo con la función de pérdida adecuada, como la entropía cruzada, y el optimizador, como Adam o SGD.

Paso 2: Entrena el modelo utilizando el conjunto de datos de entrenamiento, ajustando los pesos de la red neuronal mediante el proceso de retropropagación.

5. Evaluación del modelo

Paso 1: Evalúa el modelo utilizando el conjunto de datos de prueba y calcula métricas de rendimiento, como la precisión y el puntaje F1.

Paso 2: Analiza los resultados para determinar la efectividad del modelo en el análisis de sentimientos.

6. Clasificación de nuevos comentarios

Paso 1: Preprocesa los nuevos comentarios siguiendo los pasos de tokenización, creación de secuencias y conversión a números utilizando el vocabulario creado.





Paso 2: Utiliza el modelo entrenado para realizar predicciones en los nuevos comentarios y clasificarlos como positivos o negativos.

Paso 3: Muestra los resultados del análisis de sentimientos para cada comentario.

Tema III

Práctica 1. Clasificación de imágenes con redes neuronales convolucionales.

1. Preparación del entorno de desarrollo

Paso 1: Puedes utilizar entornos de desarrollo como Jupyter Notebook o Google Colab para escribir y ejecutar tu código.

Paso 2: Instala las bibliotecas necesarias, como TensorFlow, Keras y NumPy, utilizando el gestor de paquetes de Python, pip.

2. Preparación de los datos

Paso 1: Descarga el conjunto de datos MNIST, que contiene imágenes de dígitos escritos a mano, desde el sitio web oficial de MNIST o utilizando funciones de carga de datos disponibles en bibliotecas como Keras.

Paso 2: Carga los datos en tu entorno de desarrollo y realiza una exploración inicial para familiarizarte con ellos.

Paso 3: Preprocesa los datos, dividiéndolos en conjuntos de entrenamiento y prueba. Puedes usar la función `train_test_split` de la biblioteca scikit-learn para esto.

Paso 4: Realiza el escalado de las imágenes para normalizar los valores de píxeles. Puedes dividir los valores de píxeles entre 255 para obtener valores en el rango de 0 a 1.

3. Diseño y entrenamiento del modelo

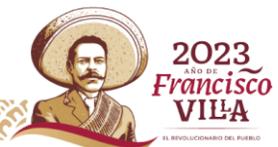
Paso 1: Define la arquitectura de la red neuronal convolucional. Puedes utilizar capas convolucionales, capas de agrupamiento (pooling) y capas completamente conectadas.

Paso 2: Compila el modelo especificando la función de pérdida y el optimizador. Para la clasificación de múltiples clases, puedes usar la función de pérdida `categorical_crossentropy` y el optimizador `adam`.

Paso 3: Entrena el modelo utilizando el conjunto de datos de entrenamiento. Ajusta el número de épocas de entrenamiento y el tamaño del lote según sea necesario.

Paso 4: Mientras entrenas el modelo, puedes monitorear la precisión y la pérdida en cada época para evaluar el rendimiento del modelo.

4. Evaluación y mejora del rendimiento





Paso 1: Evalúa el modelo utilizando el conjunto de datos de prueba. Utiliza la función `evaluate` para obtener la precisión y la pérdida del modelo en el conjunto de prueba.

Paso 2: Analiza los resultados obtenidos y, si es necesario, realiza ajustes en la arquitectura de la red neuronal convolucional. Puedes agregar más capas, ajustar los hiperparámetros o probar diferentes optimizadores para mejorar el rendimiento.

Paso 3: Realiza pruebas adicionales utilizando conjuntos de datos diferentes o imágenes reales para evaluar el desempeño del modelo en situaciones del mundo real.

Practica 2. Detección de enfermedades en aguacates utilizando redes neuronales convolucionales.

1. Preparación del entorno de desarrollo

Paso 1: Configura tu entorno de desarrollo, como Jupyter Notebook o Google Colab, e instala las bibliotecas necesarias, como TensorFlow, Keras, OpenCV y NumPy.

2. Preparación de los datos

Paso 1: Recopila un conjunto de datos que contenga imágenes de aguacates afectados por diferentes enfermedades comunes en el estado de Michoacán, como la mancha de sol, el hongo Anthracnose o la pudrición de la raíz.

Paso 2: Preprocesa los datos, asegurándote de que las imágenes estén correctamente etiquetadas con la información de las enfermedades correspondientes.

Paso 3: Divide los datos en conjuntos de entrenamiento y prueba.

3. Diseño y entrenamiento del modelo

Paso 1: Define la arquitectura de la red neuronal convolucional para la detección de enfermedades en aguacates. Puedes utilizar una arquitectura pre-entrenada como VGG16 o ResNet como punto de partida.

Paso 2: Ajusta los hiperparámetros del modelo, como el tamaño del lote, el número de épocas y la tasa de aprendizaje, según sea necesario.

Paso 3: Compila el modelo especificando la función de pérdida y el optimizador adecuados para la detección de enfermedades en aguacates.

Paso 4: Entrena el modelo utilizando el conjunto de datos de entrenamiento.

4. Evaluación y mejora del rendimiento

Paso 1: Evalúa el modelo utilizando el conjunto de datos de prueba. Calcula métricas como la precisión, el recall y el F1-score para medir la efectividad del modelo en la detección de enfermedades en aguacates.





Paso 2: Analiza los resultados obtenidos y, si es necesario, realiza ajustes en la arquitectura de la red neuronal convolucional. Puedes agregar capas adicionales, ajustar los hiperparámetros o utilizar técnicas de regularización para mejorar el rendimiento del modelo.

Paso 3: Realiza pruebas adicionales utilizando imágenes reales de aguacates afectados por enfermedades y evalúa el desempeño del modelo en situaciones del mundo real.

5. Aplicación en tiempo real

Paso 1: Utiliza el modelo entrenado para realizar la detección de enfermedades en aguacates en tiempo real. Esto implica capturar imágenes de aguacates en los huertos o en los puntos de inspección.

Paso 2: Procesa las imágenes capturadas utilizando OpenCV para detectar las regiones de interés (ROI) que podrían contener enfermedades.

Paso 3: Utiliza el modelo entrenado para clasificar las ROI y determinar la presencia y el tipo de enfermedad en cada una.

Paso 4: Visualiza los resultados obtenidos, ya sea mostrando las imágenes con las áreas detectadas o generando informes con la información de las enfermedades encontradas en los aguacates.

Practica 3. Aplicando los mismos pasos de la práctica anterior, desarrollar un sistema de reconocimiento de emociones en tiempo real utilizando redes neuronales convolucionales. El objetivo es detectar y clasificar las emociones expresadas por las personas en imágenes o secuencias de video en tiempo real.

Tema IV

Practica 1. Análisis de sentimientos en redes sociales utilizando el conjunto de datos Sentiment140. Repositorio del conjunto de datos: Stanford (<https://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip>)

1. Descarga del conjunto de datos:

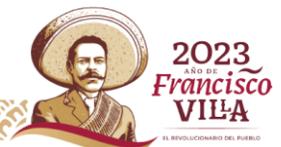
Paso 1: Accede al repositorio de conjuntos de datos de Stanford.

Paso 2: Descarga el archivo zip que contiene el conjunto de datos "Sentiment140".

2. Preprocesamiento de datos:

Paso 1: Extrae el archivo zip descargado y obtén los archivos de datos necesarios.

Paso 2: Lee los archivos y realiza el preprocesamiento necesario, como la limpieza de texto y la codificación de etiquetas de sentimiento.





3. Construcción del modelo:

Paso 1: Crea un modelo de red neuronal recurrente utilizando una capa LSTM o GRU.
Paso 2: Define la arquitectura del modelo, incluyendo el número de capas, la función de activación y otros hiperparámetros.

4. Entrenamiento del modelo:

Paso 1: Divide el conjunto de datos en conjuntos de entrenamiento y prueba.
Paso 2: Entrena el modelo utilizando los datos de entrenamiento y ajusta los pesos de las conexiones de la red neuronal recurrente.
Paso 3: Realiza ajustes en los hiperparámetros para mejorar el rendimiento del modelo.

5. Evaluación del modelo:

Paso 1: Evalúa el rendimiento del modelo utilizando los datos de prueba.
Paso 2: Calcula métricas como la precisión, la recuperación y la puntuación F1 para medir la calidad de las clasificaciones de sentimientos.

Practica 2. Predicción de cosechas utilizando redes neuronales recurrentes. Conjunto de datos: Datos históricos de producción agrícola, clima y factores ambientales.

1. Recopilación de datos:

Paso 1: Obtén datos históricos de producción agrícola de diferentes cultivos, incluyendo información sobre la cantidad de cosechas realizadas en años anteriores.
Paso 2: Obtén datos climáticos como la temperatura, la humedad, la precipitación, etc., para la misma región y período de tiempo.
Paso 3: Recopila información adicional relevante para la predicción de cosechas, como datos de fertilizantes, prácticas agrícolas, enfermedades de los cultivos, etc.

2. Preprocesamiento de datos:

Paso 1: Limpia y normaliza los datos para eliminar ruido y asegurar la consistencia.
Paso 2: Asegúrate de que los datos de producción agrícola y los datos climáticos estén alineados en términos de tiempo y ubicación geográfica.
Paso 3: Divide los datos en conjuntos de entrenamiento y prueba.

3. Construcción del modelo de red neuronal recurrente:





Paso 1: Define la arquitectura de la red neuronal recurrente, que puede incluir capas LSTM o GRU.

Paso 2: Considera la inclusión de capas adicionales, como capas densas, para mejorar la capacidad de predicción.

Paso 3: Define la función de pérdida y el algoritmo de optimización adecuados para el problema de predicción de cosechas.

4. Entrenamiento del modelo:

Paso 1: Alimenta los datos de entrenamiento a la red neuronal recurrente.

Paso 2: Ajusta los pesos de la red neuronal utilizando el algoritmo de optimización seleccionado.

Paso 3: Realiza múltiples iteraciones de entrenamiento hasta que el modelo converja o alcance un nivel de rendimiento satisfactorio.

5. Evaluación del modelo y predicción de cosechas:

Paso 1: Utiliza los datos de prueba para evaluar el rendimiento del modelo.

Paso 2: Calcula métricas de evaluación, como el error medio absoluto (MAE) o el error cuadrático medio (MSE), para medir la precisión de las predicciones.

Paso 3: Utiliza el modelo entrenado para predecir cosechas futuras en base a datos climáticos y otros factores relevantes.

Recuerda ajustar los hiperparámetros del modelo y realizar experimentos adicionales para mejorar su rendimiento. Además, considera la posibilidad de agregar técnicas de regularización y validación cruzada para evitar el sobreajuste y garantizar la generalización adecuada del modelo.

Ejemplos de conjuntos de datos etiquetados relacionados con el sector agrícola y sus respectivos repositorios:

1. Conjunto de datos: Global Maize Production

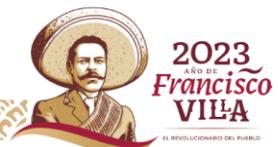
Repositorio: <https://www.kaggle.com/unitednations/global-maize-production>

2. Conjunto de datos: Global Wheat Production

Repositorio: <https://www.kaggle.com/unitednations/global-wheat-production>

3. Conjunto de datos: Crop Yield Prediction

Repositorio: <https://www.kaggle.com/vishalmane109/crop-yield-prediction>





4. Conjunto de datos: Climate Change and Agriculture

Repositorio: <https://www.kaggle.com/venky97/climate-change-and-its-effect-on-agriculture>

5. Conjunto de datos: Plant Pathology

Repositorio: <https://www.kaggle.com/c/plant-pathology-2020-fgvc7>

Estos conjuntos de datos contienen información etiquetada sobre diversos aspectos agrícolas, como la producción de maíz, trigo, predicciones de rendimiento de cultivos, cambio climático y efectos en la agricultura, y patología de plantas. Puedes acceder a ellos a través de los enlaces proporcionados y explorar los datos etiquetados disponibles para entrenar y evaluar tus modelos de redes neuronales recurrentes en el contexto agrícola.

9. Proyecto de asignatura

Fundamento:

El objetivo de este proyecto es crear una solución escalable y accesible basada en redes neuronales convolucionales para el análisis de imágenes en tiempo real. La solución permitirá a los usuarios cargar imágenes a través de una aplicación web y/o móvil, llamar al modelo en la nube a través de una API y recibir resultados de detección y clasificación de objetos.

Planeación:

Definir requisitos y alcance:

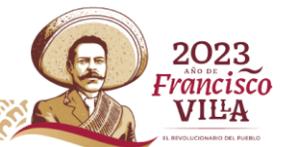
Identificar los requisitos del proyecto en términos de funcionalidad, rendimiento, seguridad y usabilidad.

Definir el alcance del proyecto, incluyendo las características clave, los casos de uso y los usuarios objetivo.

Recopilar y etiquetar conjuntos de datos:

Buscar conjuntos de datos relevantes en línea que contengan imágenes etiquetadas para entrenar y evaluar el modelo.

Etiquetar las imágenes para las clases de objetos o características de interés específicas para el proyecto.





Preprocesamiento y entrenamiento del modelo:

Realizar el preprocesamiento de los datos, como redimensionar, normalizar y dividir en conjuntos de entrenamiento y prueba.

Seleccionar y configurar el modelo de red neuronal convolucional adecuado para el problema en cuestión.

Entrenar el modelo utilizando el conjunto de datos etiquetado y optimizar los hiperparámetros para lograr un rendimiento óptimo.

Implementación del modelo en la nube:

Configurar el entorno en la nube utilizando servicios como Google Cloud, AWS o Microsoft Azure para alojar y ejecutar el modelo.

Desplegar el modelo entrenado en la infraestructura en la nube y asegurarse de que esté disponible a través de una API.

Desarrollo de la aplicación web y/o móvil:

Seleccionar un marco de desarrollo web y/o móvil adecuado, como Django, Flask, React Native o Flutter.

Diseñar e implementar la interfaz de usuario de la aplicación, incluyendo la capacidad de cargar imágenes y enviar solicitudes a la API del modelo.

Implementar la lógica de comunicación con la API, incluyendo la autenticación y el manejo de respuestas del modelo.

Pruebas y refinamiento:

Realizar pruebas exhaustivas de la aplicación para asegurarse de que funcione correctamente y cumpla con los requisitos establecidos.

Recopilar y analizar comentarios de los usuarios y realizar ajustes y mejoras según sea necesario.

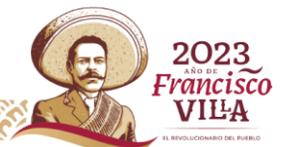
Implementación y despliegue:

Desplegar la aplicación web en un servidor web o en un servicio de alojamiento en la nube.

Para la aplicación móvil, compilar el código en un archivo ejecutable y publicarlo en las tiendas de aplicaciones correspondientes.

Desarrollo:

1. Definir requisitos y alcance:





- Identificar los requisitos funcionales y no funcionales del proyecto en términos de funcionalidad, rendimiento, seguridad y usabilidad.
- Determinar el alcance del proyecto, incluyendo las características clave, los casos de uso y los usuarios objetivo.

2. Recopilar y etiquetar conjuntos de datos:

- Buscar conjuntos de datos relevantes en línea que contengan imágenes etiquetadas para entrenar y evaluar el modelo.
- Etiquetar las imágenes para las clases de objetos o características de interés específicas para el proyecto. Puedes utilizar herramientas de etiquetado como LabelImg o RectLabel.

3. Preprocesamiento y entrenamiento del modelo:

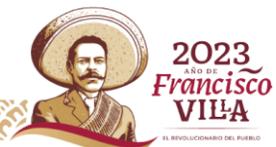
- Realizar el preprocesamiento de los datos, como redimensionar, normalizar y dividir en conjuntos de entrenamiento y prueba.
- Seleccionar y configurar el modelo de red neuronal convolucional adecuado para el problema en cuestión. Puedes utilizar bibliotecas como TensorFlow o PyTorch.
- Entrenar el modelo utilizando el conjunto de datos etiquetado y optimizar los hiperparámetros para lograr un rendimiento óptimo. Puedes utilizar técnicas de transfer learning si es necesario.

4. Implementación del modelo en la nube:

- Configurar el entorno en la nube utilizando servicios como Google Cloud, AWS o Microsoft Azure para alojar y ejecutar el modelo. Puedes utilizar servicios como Google Cloud AI Platform, AWS SageMaker o Azure Machine Learning.
- Desplegar el modelo entrenado en la infraestructura en la nube y asegurarse de que esté disponible a través de una API. Puedes utilizar frameworks como Flask o Django para crear la API.

5. Desarrollo de la aplicación web y/o móvil:

- Seleccionar un marco de desarrollo web y/o móvil adecuado, como Django, Flask, React Native o Flutter.
- Diseñar e implementar la interfaz de usuario de la aplicación, incluyendo la capacidad de cargar imágenes y enviar solicitudes a la API del modelo. Puedes utilizar bibliotecas como React o Vue.js para el desarrollo web, y Flutter o React Native para el desarrollo móvil.
- Implementar la lógica de comunicación con la API, incluyendo la autenticación y el manejo de respuestas del modelo. Puedes utilizar bibliotecas como Axios para realizar solicitudes HTTP.





6. Pruebas y refinamiento:

- Realizar pruebas exhaustivas de la aplicación para asegurarse de que funcione correctamente y cumpla con los requisitos establecidos. Puedes utilizar herramientas de pruebas como Jest, Pytest o Selenium.
- Recopilar y analizar comentarios de los usuarios y realizar ajustes y mejoras según sea necesario. Puedes utilizar herramientas de seguimiento de problemas como Jira o Trello para gestionar los problemas y solicitudes de los usuarios.

7. Implementación y despliegue:

- Desplegar la aplicación web en un servidor web o en un servicio de alojamiento en la nube. Puedes utilizar servicios como Heroku, Netlify o AWS S3.
- Para la aplicación móvil, compilar el código en un archivo ejecutable y público.

Evaluación

La rúbrica contemplará los siguientes criterios:

Planeación 20%

Desarrollo 40%

Publicación 10%

Exposición del proyecto 20%

Trabajo en equipo 10%

10. Evaluación por competencias

La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño encada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Rúbricas o productos, señalados en cada unidad académica dentro de las actividades de aprendizaje.
- Prácticas propuestas y su presentación y exposición en plenaria. Algunas se evaluarán por equipo.
- Portafolio de evidencias de Información recabada durante las consultas e investigaciones solicitadas, plasmadas en documentos escritos.





- Descripción de otras experiencias concretas que se obtendrán al participar en discusiones, exposiciones o cualquier otro medio didáctico-profesional que trate sobre la materia y que deberán realizarse durante el curso académico.
- Exámenes teórico-prácticos para comprobar la efectividad del estudiante en la resolución de casos prácticos.

11. Fuentes de información

1. Santanu Pattanayak. Pro Deep Learning with Tensorflow. Apress. 2017
2. Tom M. Mitchell. Machine Learning. McGraw-Hill. 1997.
3. Joseph Howse. OpenCV Computer Vision with Python. BIRMINGHAM – MUMBAI. 2013.
4. Aurélien Géron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O’Reilly Media.
5. Adrian Rosebrock. Deep Learning for computer vision with python. PyImageSearch. 2017
6. Richard Szeliski. Computer Vision: Algorithms and Applications. Springer. 2010.
7. Adrian Rosebrock. Python for computer vision with OpenCV and Deep Learning. PyImageSearch. 2021
8. Ian Goodfellow, Yoshua Bengio, and Aaron Corville. Deep Learning. MIT Press. 2016.
9. Mohamed Elgendy. Deep Learning for vision Systems. Manning Publications. 2020





1. Datos Generales de la asignatura

Nombre de la asignatura:	Desarrollo de Aplicaciones Web.
Clave de la asignatura:	DAD-2404
SATCA¹:	2-3-5
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las competencias profesionales:

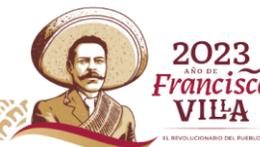
- Aplicar conocimientos científicos y tecnológicos en la solución de problemas en el área de sistemas computacionales con un enfoque interdisciplinario.
- Administrar las tecnologías de la información, para estructurar proyectos estratégicos.
- Analizar, modelar, desarrollar, implementar y administrar sistemas de información web para aumentar la productividad y competitividad de las organizaciones.

La importancia de esta asignatura es que permite al alumno desarrollar e implementar Aplicaciones Web, que den soluciones a las necesidades productivas y comerciales de las empresas, con persistencia y reporte de información, incorporando APIS o microservicios de autenticación, autorización, rol, carrito de compra, pasarela de pagos, chatbot, inclusive con características de IA.

Intención didáctica

La asignatura se orienta a ser práctica y capaz de desarrollar en el estudiante la habilidad para el desarrollo de proyectos web, el proceso de producción y despliegue de los mismos.

¹ Sistema de Asignación y Transferencia de Créditos Académicos





- En el tema I. El estudiante será capaz de Identificar los marcos de trabajo actuales que permiten el desarrollo de aplicaciones Web, su objetivo, los elementos y herramientas disponibles e identificar la arquitectura que distingue a cada uno de ellos.
- En el tema II. El estudiante será capaz emplear los marcos de trabajo y sus herramientas de ventanas, controles y métodos, para implementar la interfaz gráfica de un proyecto web.
- En el tema III. El estudiante será capaz de implementar un modelo de persistencia de datos relacional o no relacional a la interfaz gráfica del proyecto web a nivel funcional.
- En el tema IV. El estudiante será capaz de diseñar e implementar el plan de seguridad para el proyecto web a nivel de servidores, de plataforma, de aplicación y a la conexión y comunicación.
- En el tema V. El estudiante será capaz de integrar servicios a la aplicación web de autenticación y autorización; carritos de compra y pasarelas de pago; además de la generación de reportes requeridos.
- En el tema VI. El estudiante será capaz de gestionar servidores de despliegue, su seguridad y el escalamiento para cargar el proyecto web y llevar a cabo la producción del mismo en la web.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Jiquilpan Jiquilpan Michoacán, 08 de mayo de 2023	M. E. Luis Irepan Núñez Ing. Leonardo Martínez González M. I. S. C. Erick Dalet Villanueva Mascort Lic. Martha Elizabeth González Zepeda Ing. Fernando Carranza Campos	Diseño y desarrollo de los programas de estudio del módulo de especialidad para la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura





- Desarrollar e implementar un proyecto web a partir de la elección de un marco de trabajo, con acceso a dato y la incorporación de servicios.
- Identificar y elegir el marco de trabajo front-end y back-end más adecuado para el proyecto web.
- Incorporar el diseño UI al proyecto web.
- Implementar bases de datos relacionales y no relacionales en un proyecto web.
- Identificar los requisitos de seguridad del cliente (o del proyecto web), diseñar e implementar un plan de seguridad.
- Capacidad para incorporar e implementar autenticación, roles, carrito de compra, pasarelas de pagos, servicios de IA y emisión de reportes a proyectos web.
- Determinar y configurar el modelo de despliegue y producción, considerando la seguridad y el escalamiento.

5. Competencias previas

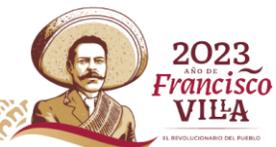
Esta asignatura es la aplicación en un desarrollo web de los conocimientos en: marcos de trabajo, diseño de la interfaz, modelado de datos, seguridad, servicios y la configuración de la producción y despliegue.

Esta asignatura tiene relación previa con las signaturas:

- Diseño e de interfaz y experiencia de usuario,
- Arquitectura orientada a servicios,
- Visión por computadora y Machine Learning.

Considerando las asignaturas anteriormente mencionadas se consideran las siguientes competencias específicas:

- Aplicar el diseño de interfaz de usuario en un proyecto web.
- Identificar, desarrollar microservicios.
- Implementar servicios de inteligencia artificial como procesamiento y análisis de datos, deep learning, redes neuronales proporcionados en la web en el proyecto a desarrollar.





Tomar decisiones con base en los elementos teórico-práctico adquiridos que permitan generar aplicaciones web integrales y complejas que incluyan la seguridad requerida, el acceso a datos, la incorporación de servicios de terceros instaladas en y configuradas en servidores.

6. Temario

No.	Temas	Subtemas
1	Marcos de Trabajo.	1.1. Análisis y selección de un marco de trabajo. 1.2. Funciones avanzadas.
2	Interfaces Gráficas (GUI).	2.1. Ventanas. 2.2. Controles. 2.3. Métodos.
3	Persistencia de Datos.	3.1. Bases de datos relacionales. 3.2. Operaciones de bases de datos relacionales. 3.3. Bases de datos no relacionales. 3.4. Operaciones de bases de datos no relacionales.
4	Seguridad.	4.1. Introducción. 4.2. Implementación de plan de seguridad.
5	Servicios.	5.1. Reportes. 5.2. Pasarela de pagos y carrito de compras. 5.3. Autenticación, roles y autorización. 5.4. Servicios de IA.





6	Despliegue.	<p>6.1. Gestión de servicios de despliegue.</p> <p>6.2. Modelos y estándares para despliegue.</p> <p>6.3. Seguridad.</p> <p>6.4. Escalamiento.</p>
---	-------------	--

7. Actividades de aprendizaje de los temas

I Marcos de Trabajo.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Comprende, investiga y selecciona el marco de trabajo más adecuados para desarrollar el front end y el back end. <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidad para buscar y analizar información proveniente de fuentes diversas Capacidad de análisis y síntesis. Habilidad en el manejo de equipo de cómputo. Capacidad para trabajar en equipo. Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> Investiga los marcos de trabajos para el proyecto web en el back-end y front-end, identifica la tecnología empleada, las ventajas, desventajas y el tipo de desarrollo.
II Interfaces Gráficas (GUI).	
Competencias	Actividades de aprendizaje





<p>Específica(s):</p> <ul style="list-style-type: none"> Habilidad para aplicar las herramientas gráficas del marco de trabajo en el diseño de una interfaz GUI. Capacidad para evaluar el diseño de la GUI aplicando UX. <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidad para buscar y analizar información de fuentes diversas. Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Solución de problemas. Capacidad crítica y autocrítica. Capacidad para trabajo en equipo. Compromiso ético. Capacidad de generar nuevas ideas (creatividad). 	<ul style="list-style-type: none"> Diseñar una interfaz de usuario evaluando la experiencia de usuario. Implementar el diseño de interfaz en el marco de trabajo de acuerdo con la naturaleza del desarrollo web.
III Persistencia de Datos.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Capacidad para implementar la persistencia de datos en la web. <p>Genéricas:</p>	<ul style="list-style-type: none"> Diseñar la base de datos de acuerdo al proyecto web a desarrollar. Despliegue de la base de datos en la nube. Crear las conexiones hacia la base de datos.



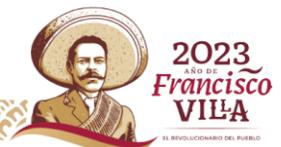


<ul style="list-style-type: none"> Habilidad para buscar y analizar información de fuentes diversas. Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Solución de problemas. Capacidad crítica y autocrítica. Capacidad para trabajo en equipo. Compromiso ético. Capacidad de generar nuevas ideas (creatividad). 	
IV Seguridad.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Implementar un plan de seguridad en el proyecto web. <p>Genéricas:</p> <ul style="list-style-type: none"> Habilidad para buscar y analizar información de fuentes diversas. Capacidad de análisis y síntesis. Capacidad de organizar y planificar. Comunicación oral y escrita. Solución de problemas. Trabajo en equipo. Capacidad de aplicar los conocimientos. Capacidad de generar nuevas ideas (Creatividad). Capacidad para diseñar y gestionar proyectos. 	<ul style="list-style-type: none"> Establecer el plan de seguridad a nivel de plataforma, aplicación, comunicación y servidores de acuerdo al desarrollo web.





<ul style="list-style-type: none"> Búsqueda del logro. Liderazgo. 	
V Servicios.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Integración de servicios en el proyecto web. <p>Genéricas:</p> <ul style="list-style-type: none"> Capacidad de análisis y síntesis. Habilidad para buscar y analizar información proveniente de fuentes diversas. Solución de problemas. Capacidad de aplicar los conocimientos en la práctica. Capacidad de aprender. Habilidad para trabajar de forma autónoma y en equipo. Búsqueda del logro. Liderazgo. 	<ul style="list-style-type: none"> Investigar los servicios existentes en la web, como autenticación, autorización por terceros, servicios de IA, reconocimiento facial, análisis y predicción de datos, chatbots, pasarelas de pago, geolocalización, etc. Exponer en equipo los servicios investigados. Implementar los servicios en el desarrollo web propuesto.
VI Despliegue.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p>	<ul style="list-style-type: none"> Investigar las plataformas web para despliegue de desarrollo web. Cargar el desarrollo web a la plataforma de despliegue.





<ul style="list-style-type: none"> • Capacidad para configurar el despliegue del desarrollo web en el servidor seleccionado. • Capacidad para poner en producción un desarrollo web. <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidad para buscar y analizar información de fuentes diversas. • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Solución de problemas. • Trabajo en equipo. • Capacidad de comunicación interdisciplinaria. • Capacidad de aplicar los conocimientos. • Capacidad de generar nuevas ideas (creatividad). • Capacidad para diseñar y gestionar proyectos. • Búsqueda del logro. • Liderazgo. 	<ul style="list-style-type: none"> • Evaluar el despliegue del desarrollo web.
---	---

8. Práctica(s)

<p>Tema I Práctica 1. Implementar el modelo marco de trabajo para el proyecto de web.</p> <p>Tema II Práctica 1. Implementar el framework de acuerdo al diseño de la UI del proyecto web.</p> <p>Tema III</p>





Práctica 1. Creación e Implementación de la base de datos en la nube.
Práctica 2. Realizar y probar las conexiones a la base de datos.

Tema IV

Práctica 1. Diseñar e Incorporar el plan de seguridad al proyecto web.

Tema V

Práctica 1. Definir e Integrar los servicios que requiera el proyecto web.

Tema V

Práctica 1. Colocar en producción el proyecto web y realizar las pruebas de ejecución y desempeño.

9. Proyecto de asignatura

Fundamento:

El Objetivo es el desarrollo de proyecto web que se encuentre diseñado considerando los fundamentos de UI y UX, integrar el manejo de datos mediante base de datos relacionales y no relacionales, utilizar servicios de inteligencia artificial, aplicar el plan de seguridad y realizar el despliegue de la aplicación en la nube. Los proyectos a realizarse se establecen a partir de especificaciones que indique el docente o necesidades que los estudiantes determinen que existen en el entorno local, regional, nacional e internacional, factibles del diseño de la aplicación web.

La aplicación puede ser la automatización de procesos administrativos y empresariales, con el despliegue y consumo de modelos de servicios de IA.

Planeación:

Los equipos serán de 2 a 3 integrantes, el proyecto a realizar podrá considerar en su desarrollo las competencias adquiridas en asignaturas relacionadas con la ingeniería del software; así como la aplicación de los principios de ingeniería del software en las etapas de análisis y diseño de un sistema de información y establecer para el plan del proyecto, y dar solución considerando la complejidad de cada uno de los proyectos.





Las actividades de la determinación, especificación de requerimientos, que sirven además de base para documentar el proyecto y tener un registro y control para la implementación de la solución. El siguiente paso sería la implementación de la solución y finalmente la implantación y publicación en la Internet.

Desarrollo:

En el desarrollo se van a llevar a cabo las actividades planteadas en la planeación, pero se detallan de forma más específica otras actividades particulares para su implementación.

La aplicación web debe considerar:

- Un diseño de acuerdo a las políticas de la empresa a la que se le desarrolle, Enel aspecto de interfaz de usuario y de datos.
- Un diseño equilibrado aplicando los fundamentos de la UI y evaluado a partir de la UX.
- La persistencia de datos mediante el uso de una base de datos relacional o no relacional.
- Servicios de seguridad como: Roles de usuarios, autenticación y validación de usuarios, autorización, rastros de auditoría; incorporando servicios de IA.
- Servicios de inteligencia artificial en el aspecto comercial como: carritos de compra y preferencia de compra, pasarelas de pago, etc.
- La especificación e implementación de un plan de seguridad para la plataforma, aplicación y servidores.
- La publicación de la aplicación web en un servidor en la nube.
- La documentación de la aplicación.

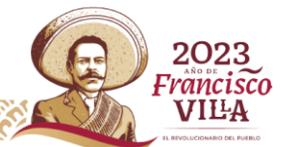
Evaluación

La rúbrica contemplará los siguientes criterios:

- Planeación 20%
- Desarrollo 40%
- Publicación 10%
- Exposición del proyecto 20%
- Trabajo en equipo 10%

10. Evaluación por competencias

[Empty box for evaluation by competencies]





La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño encada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Rúbricas o productos, señalados en cada unidad académica dentro de las actividades de aprendizaje.
- Prácticas propuestas y su presentación y exposición en plenaria. Algunas se evaluarán por equipo.
- Portafolio de evidencias de Información recabada durante las consultas e investigaciones solicitadas, plasmadas en documentos escritos.
- Descripción de otras experiencias concretas que se obtendrán al participar en discusiones, exposiciones o cualquier otro medio didáctico-profesional que trate sobre la materia y que deberán realizarse durante el curso académico.
- Exámenes teórico-prácticos para comprobar la efectividad del estudiante en la resolución de casos prácticos.

11. Fuentes de información

1. Santana Roldán, C. (2021). Ejercicios Prácticos con React (1ª ed.). Alfaomega, Marcombo.
2. Vázquez Olivares, J. A., & Lorenzo Martínez, D. (2020). Ruby on Rails (1ª ed.). Alfaomega, Marcombo.
3. Boada Oriols, M., & Gómez Gutiérrez, J. A. (2019). El Gran Libro de Angular (1ª ed.). Alfaomega, Marcombo.
4. Melé, A. (2020). Django 2. (1ª ed.) Alfaomega, Marcombo.
5. Bradshaw, S., Brazil, E., & Chodorow, K. (Eds.). (2019). MongoDB: The Definitive Guide: Powerful and Scalable Data Storage (3rd ed.)

Referencias:

Documentación oficial: <https://reactjs.org/>

Documentación oficial: <https://angular.io/>

Documentación oficial: <https://docs.djangoproject.com/>

Documentación oficial: <https://rubyonrails.org/>

Tutorial Ruby on Rails de RailsGuides:

https://guides.rubyonrails.org/getting_started.html





1. Datos Generales de la asignatura

Nombre de la asignatura:	Desarrollo de Aplicaciones Móviles.
Clave de la asignatura:	DAD-2405
SATCA¹:	2-3-5
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura
<p>La tendencia del uso de dispositivos móviles presenta un campo que debe ser aprovechado.</p> <p>Esta asignatura aporta al perfil del egresado la capacidad de desarrollar aplicaciones para dispositivos móviles.</p> <p>La asignatura es posterior a las materias que se encuentran definidas en programación e ingeniería y tratamiento de información ya que éstas permiten definir el desarrollo de soluciones para dispositivos móviles.</p>
Intención didáctica
<p>La asignatura debe ser práctica y capaz de despertar en el estudiante la habilidad para desarrollar aplicaciones móviles, además cubre con la necesidad inminente que tiene un ingeniero al enfrentarse al uso cotidiano de tecnologías tales como: telefonía celular, tabletas y demás dispositivos móviles que permitan una comunicación efectiva y versátil.</p> <p>El temario está organizado en seis temas. El primer tema, introduce al estudiante a los conceptos y evolución de los dispositivos móviles.</p> <p>El tema dos, se centra en el aprendizaje de un lenguaje de programación móvil aplicando una</p>

¹ Sistema de Asignación y Transferencia de Créditos Académicos





metodología de programación orientada a objetos.

El tema tres, permite la aplicación de diferentes componentes y controles en el desarrollo de aplicaciones móviles con interfaces de usuario amigables e intuitivas.

En el tema cuatro, el alumno aplica componentes avanzados en la construcción de aplicaciones que requieren el uso de mapas, sensores, hilos, entre otros temas.

El tema cinco se enfoca en la aplicación de elementos utilizados para el acceso y presentación de datos en dispositivos móviles.

Por último, el tema seis se centra en el despliegue de las aplicaciones móviles en la web.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Jiquilpan Jiquilpan Michoacán, 08 de mayo de 2023	Ing. Erik Dalet Villanueva Mascort Lic. Martha Elizabeth González Zepeda Lic. Armando Hernández Gálvez	Diseño y desarrollo de los programas de estudio del módulo de especialidad para la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<ul style="list-style-type: none"> • Identifica los elementos que intervienen en el desarrollo móvil. • Utiliza y maneja entornos de desarrollo para dispositivos móviles. • Utiliza técnicas de modelado para la solución de problemas. • Aplica un lenguaje para la solución de problemas para dispositivos móviles. • Aplica algoritmos de inteligencia artificial en la solución de problemas.





5. Competencias previas

- Analiza y soluciona problemas informáticos y representa su solución mediante herramientas de software orientado a objetos.
- Identifica y analiza necesidades de información para su representación, tratamiento y automatización para la toma de decisiones.
- Diseña esquemas de base de datos para generar soluciones al tratamiento de información.
- Identifica y aplica distintos SGBD, así como sus herramientas.
- Aplica el entorno operativo del SGBD, para crear el esquema de una base de datos.
- Aplica esquema de integridad, seguridad y recuperación.

6. Temario

No.	Temas	Subtemas
1	Introducción a la Programación Móvil.	1.1. Dispositivos móviles. 1.1.1. Conceptualización. 1.1.2. Clasificación. 1.2. Sistemas operativos móviles. 1.2.1. Características. 1.2.2. Costo desarrollador. 1.3. Tecnologías móviles. 1.3.1. Descripción. 1.3.2. Objetivo. 1.3.3. Movilidad. 1.3.4. Redes móviles. 1.4. Entornos de desarrollo y lenguajes de programación móvil. 1.5. Aplicaciones móviles. 1.5.1. Conceptualización. 1.5.2. Tipos de aplicaciones móviles. 1.5.3. Objetivo de desarrollo. 1.5.4. Decisión de desarrollo. 1.5.5. Mercado de aplicaciones móviles.





		<p>1.5.6. Publicación de aplicaciones.</p> <p>1.6. Emuladores.</p>
2	Lenguaje de Programación Móvil.	<p>2.1. Tipos de datos.</p> <p>2.2. Variables, constantes y operadores.</p> <p>2.3. Cadenas de texto.</p> <p>2.4. Sentencias de control.</p> <p>2.4.1. if.</p> <p>2.4.2. while.</p> <p>2.4.3. for.</p> <p>2.4.4. switch.</p> <p>2.5. Arreglos.</p> <p>2.6. Funciones.</p> <p>2.7. Colecciones.</p> <p>2.8. Programación orientada a objetos.</p> <p>2.8.1. Estructura.</p> <p>2.8.2. Propiedades.</p> <p>2.8.3. Setters y Getters.</p> <p>2.8.4. Métodos.</p> <p>2.8.5. Clases.</p> <p>2.8.6. Herencia.</p> <p>2.8.7. Enumeraciones.</p> <p>2.8.8. Protocolos.</p>
3	Desarrollo de Aplicaciones Móviles.	<p>3.1. Metodología, desarrollo y ejecución.</p> <p>3.2. AutoLayout y vistas.</p> <p>3.3. Uso de componentes y controles.</p> <p>3.3.1. Botones.</p> <p>3.3.2. Etiquetas.</p> <p>3.3.3. Componentes de texto.</p> <p>3.3.4. Componentes de selección.</p> <p>3.3.5. Barras de desplazamiento.</p> <p>3.3.6. Listas.</p> <p>3.3.7. Componentes de fecha.</p> <p>3.3.8. Menús, Etc.</p>





		3.4. Creación de interfaces. 3.5. Formularios.
4	Elementos Avanzados de Desarrollo Móvil.	4.1. Clases adaptadoras. 4.2. Hilos. 4.2.1. Síncronos. 4.2.2. Asíncronos. 4.3. Mapas. 4.4. Manejo Web. 4.5. Sensores.
5	Administración de Datos en Dispositivos Móviles.	5.1. Introducción. 5.2. Manejo de datos de forma persistente. 5.3. Manipulación de datos. 5.4. Persistencia usando base de datos.
6	Despliegue de Aplicaciones Móviles.	6.1. Gestión de servicios de despliegue. 6.2. Modelos y estándares para despliegue. 6.3. Seguridad. 6.4. Escalamiento.

7. Actividades de aprendizaje de los temas

I Introducción a la Programación Móvil.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Identificar los elementos requeridos en el desarrollo móvil. 	<ul style="list-style-type: none"> Realiza una investigación de los aspectos que se deben considerar para desarrollar una aplicación móvil. Realizar un mapa conceptual de los elementos básicos del desarrollo móvil.



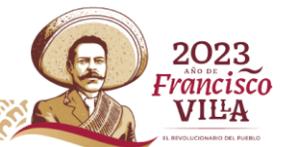


<ul style="list-style-type: none"> • Analizar las diferentes alternativas de desarrollo móvil y requisitos necesarios. • Reconoce e identifica los diferentes tipos de aplicaciones que se pueden desarrollar. • Conocer el proceso que se requiere para publicar una aplicación. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Comunicación oral y escrita. 	<ul style="list-style-type: none"> • Hacer una tabla comparativa de las diferentes tecnologías de desarrollo móvil. • Realiza un comparativo de ventajas y desventajas de los diferentes tipos de aplicaciones. • Investiga y presenta en un diagrama de flujo de procesos, la secuencia de actividades que se necesita para publicar una aplicación.
II Lenguaje de Programación Móvil.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Identificar las diferencias de las instrucciones del lenguaje de programación móvil, con respecto a otros lenguajes de programación. • Conocer y aplicar los elementos e instrucciones del lenguaje a la solución de problemas reales. • Aplicar los conceptos de clases y objetos en el desarrollo de programas para la solución de problemas de 	<ul style="list-style-type: none"> • Investiga sobre la sintaxis, funcionamiento y aplicación de los elementos e instrucciones del lenguaje y realiza un cuadro comparativo con respecto a las instrucciones de otro lenguaje. • Ejemplifica en programas que resuelvan problemas reales, discutirlo en grupo. • Realiza ejercicios prácticos, donde se utilicen las diferentes estructuras desarrollando algoritmos y programas, presentarlos y discutirlos en grupo. • Crea clases que reúnan los datos miembro necesarios para resolver un problema y así implementar el encapsulamiento.



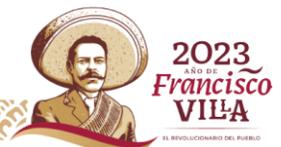


<p>acuerdo al paradigma orientado a objetos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Capacidad de aplicar los conocimientos en la práctica. • Toma de decisiones. 	
III Desarrollo de Aplicaciones Móviles.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Utiliza técnicas de modelado para la solución de problemas. • Aplicar la sintaxis de un lenguaje para realizar interfaces de usuario utilizando componentes. • Realizar solución de problemas para dispositivos móviles que integren manejo de eventos e interfaces adaptables a diferentes dispositivos. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. 	<ul style="list-style-type: none"> • Enumera el proceso de ejecución de una aplicación móvil. • Desarrolla aplicaciones que se adapten a diferentes resoluciones. • Desarrolla aplicaciones que integren componentes y eventos. • Desarrolla aplicaciones para interactuar con el usuario de una manera amigable, utilizando una interfaz gráfica de usuario manipuladas a través de eventos. • Desarrolla aplicaciones para dispositivos móviles, considerando su entorno operativo.





<ul style="list-style-type: none"> • Toma de decisiones. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. • Capacidad de aplicar los conocimientos en la práctica. 	
IV Elementos Avanzados de Desarrollo Móvil.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Utiliza técnicas de modelado para la solución de problemas. • Aplicar la sintaxis de un lenguaje para realizar aplicaciones que utilicen sensores. • Realizar solución de problemas para dispositivos móviles donde se apliquen elementos web, hilos o mapas. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. 	<ul style="list-style-type: none"> • Desarrolla e implementa aplicaciones que utilicen algún tipo de conectividad (Wi-Fi, Bluetooth, Etc.) presentarla en plenarias. • Desarrolla e implementa aplicaciones que utilicen algún tipo de sensor (acelerómetro, giroscopio, luz, Etc.). • Desarrolla e implementa aplicaciones que requieren el uso de hilos. • Desarrolla e implementa aplicaciones donde se vea el uso de mapas y ubicación.





<ul style="list-style-type: none"> • Capacidad de aplicar los conocimientos en la práctica. 	
<p>V Administración de Datos en Dispositivos Móviles.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <ul style="list-style-type: none"> • Desarrollar aplicaciones que utilicen el manejo de datos, considerando su entorno operativo. • Implementar soluciones sobre dispositivos móviles que realicen manipulación de datos utilizando una base de datos. • Aplicar tecnologías de conectividad de base de datos actuales y emergentes. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Habilidades interpersonales. • Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> • Desarrolla aplicaciones en dispositivos móviles que tengan acceso, manipulación y visualización de datos, elaborar la documentación. • Realiza aplicaciones que generen recuperación de datos mediante controles. • Realiza aplicaciones que puedan implementar el CRUD (create, read, update y delete).





VI Despliegue de Aplicaciones Móviles.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Implementar la configuración del despliegue en el servidor seleccionado. • Realizar la puesta en producción de la aplicación móvil. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Trabajo en equipo. • Capacidad de comunicación interdisciplinaria. • Capacidad de aplicar los conocimientos en la práctica. • Liderazgo. 	<ul style="list-style-type: none"> • Investiga las plataformas web que permitan el despliegue de aplicaciones móviles. • Enumera el proceso de despliegue de una aplicación móvil. • Carga la aplicación móvil a la plataforma de despliegue. • Evalúa el despliegue de la aplicación móvil.

8. Práctica(s)

<p>Tema I</p> <p>Práctica 1. Descubriendo el entorno de desarrollo.</p> <p>Práctica 2. Analizando las ventajas y desventajas de los diferentes tipos de aplicaciones.</p>





Tema II

- Práctica 1. Desarrollar programas que hagan uso de los elementos del lenguaje de programación móvil.
- Práctica 2. Modelar clases y objetos que puedan utilizarse en aplicaciones a través del lenguaje de programación móvil.

Tema III

- Práctica 1. Editar, compilar y ejecutar distintos programas que incluyan el uso de formulario, botones, etiquetas y cajas de texto.
- Práctica 2. Desarrolla aplicaciones que incluyan la programación de eventos.
- Práctica 3. Desarrollar aplicaciones que incluyan la generación de nuevos eventos.
- Práctica 4. Desarrollar aplicaciones que manejen librerías y componentes proporcionados por el IDE.

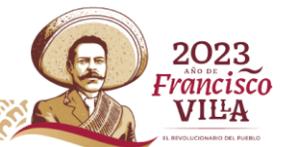
Tema IV

- Práctica 1. Desarrollar programas que implementen el uso de hilos y manipulen sus distintos estados.
- Práctica 2. Desarrollar programas que implementen el uso de varios hilos que compartan recursos y estén sincronizados.
- Práctica 3. Desarrollar componentes visuales o no visuales a partir de requerimientos previamente definidos y aplicarlos en el diseño de aplicaciones.
- Práctica 4. Desarrollar aplicaciones móviles donde se utilicen diferentes tipos de sensores y la IA.
- Práctica 5. Desarrollar una aplicación con programación concurrente que resuelva un problema real.
- Práctica 6. Desarrollar aplicaciones donde sea necesario el uso de mapas o web.

Tema V

- Práctica 1. Desarrollar una aplicación que permita establecer la conexión a un origen de datos.
- Práctica 2. Desarrollar aplicaciones donde utilice el envío, recepción y visualización de datos.
- Práctica 3. Desarrollar un juego para un dispositivo móvil.
- Práctica 4. Desarrollar aplicaciones donde se utilice algún gestor de base de datos.

Tema V





- Práctica 1. Analizar las ventajas y desventajas de al menos tres servidores de despliegue.
- Práctica 2. Desarrollar la configuración según el modelo de despliegue, garantizando la seguridad de la aplicación móvil.
- Práctica 3. Puesta en producción de la aplicación móvil.
- Práctica 4. Analizar el futuro escalamiento de la aplicación móvil.

9. Proyecto de asignatura

Fundamento:

El objetivo del proyecto es desarrollar en equipo una propuesta de una aplicación móvil que dé solución a un problema real o proceso que requiera la aplicación de un programa móvil. Los proyectos a realizar se establecen a partir de especificaciones que realice el docente o necesidades que los alumnos determinen que existen en el entorno.

La aplicación puede incluir componentes generadores de eventos, construyendo interfaces amigables, utilizando sensores, programación concurrente o el uso del manejo de datos de forma persistente.

Planeación:

Los equipos serán de tres integrantes como máximo, el proyecto a realizar podrá considerar en su desarrollo las competencias adquiridas en asignaturas relacionadas con el desarrollo de software; así como los principios del desarrollo ágil. Sin olvidar tanto elementos de seguridad como de verificación de la calidad del proyecto.

Desarrollo:

Durante el desarrollo se llevarán a cabo las actividades planteadas en la planeación y a las necesidades del proyecto, pero se desarrollan de forma particular algunas propuestas que deben de cumplirse.

La aplicación debe contemplar:

- Hacer la aplicación funcional.
- El diseño de la aplicación deberá ser acorde a las políticas de la empresa que recibirá el beneficio.
- Que los diferentes integrantes en el desarrollo del proyecto participen en él.
- Implementar técnicas de UI y UX para mejorar la experiencia de usuario.





- Diseño de pruebas de funcionamiento.
- Analizar la posibilidad de publicación de la aplicación en el market.
- Implementar servicios de IA como (autenticación mediante el reconocimiento facial, búsquedas inteligentes, chatbots, mejorar el servicio al cliente mediante algoritmos inteligentes).
- Documentación de la aplicación.

Evaluación

La rúbrica contemplará los siguientes criterios:

Planeación 20%

Desarrollo 40%

Publicación 10%

Exposición del proyecto 20%

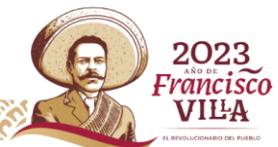
Trabajo en equipo 10%

10. Evaluación por competencias

La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño encada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Rúbricas o productos, señalados en cada unidad académica dentro de las actividades de aprendizaje.
- Prácticas propuestas y su presentación y exposición en plenaria. Algunas se evaluarán por equipo.
- Portafolio de evidencias de Información recabada durante las consultas e investigaciones solicitadas, plasmadas en documentos escritos.
- Descripción de otras experiencias concretas que se obtendrán al participar en discusiones, exposiciones o cualquier otro medio didáctico-profesional que trate sobre la materia y que deberán realizarse durante el curso académico.
- Exámenes teórico-prácticos para comprobar la efectividad del estudiante en la resolución de casos prácticos.

11. Fuentes de información





1. Becerril, S. (2017) Swift Aprende a crear Apps para iPhone y iPad. México: Alfaomega.
2. Flores, E. (2018) Aprende a programar Swift - Programación iOS: Tercera Edición. España: It Campus Academy
3. Ribas, J. (2016) Desarrollo de aplicaciones para Android. Edición 2018. España: Anaya Multimedia.
4. Phillips, B. y Stewart, C. Programación con Android. España: Marcombo.
5. Amaro, J. (2011) Programación de dispositivos móviles a través de ejemplos. España: Marcombo.
6. Darcey, L. y Conder, S. (2012) Android 4, España: Anaya Multimedia.
7. Fanlo, J. (2012). J2ME y Android. España Seas.
8. Firtman, M. (2012) JQuery mobile: Aplicaciones HTML5 para móviles. Madrid: Anaya Multimedia.
9. Lee, W. (2013). Adroid, desarrollo de aplicaciones ganadoras. España: Anaya Multimedia.
10. Rodger, R. (2012). Desarrollo de aplicaciones en la nube para dispositivos móviles. España: Anaya Multimedia.
11. San Juan, C. (2012). Programación multimedia y dispositivos móviles. España: Garceta.
12. Zecher, M (2011). Desarrollo de juegos para Android. España: Anaya Multimedia.

Referencia

<https://www.apple.com/mx/swift/> (08 de mayo de 2023)

