

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL

PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

- 1. Unidad Académica:** Facultad de Ingeniería, Arquitectura y Diseño, Ensenada; Facultad Ciencias Químicas e Ingeniería, Tijuana; y Facultad de Ciencias de la Ingeniería y Tecnología, Valle de las Palmas
- 2. Programa Educativo:** Ingeniero en Software y Tecnologías Emergentes
- 3. Plan de Estudios:** 2022-1
- 4. Nombre de la Unidad de Aprendizaje:** Análisis de Tecnologías Emergentes
- 5. Clave:** 40009
- 6. HC:** 02 **HT:** 00 **HL:** 02 **HPC:** 00 **HCL:** 00 **HE:** 02 **CR:** 06
- 7. Etapa de Formación a la que Pertenece:** Disciplinaria
- 8. Carácter de la Unidad de Aprendizaje:** Obligatoria
- 9. Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



Equipo de diseño de PUA

Bernabé Rodríguez Tapia
Juan Iván Nieto Hipólito
Juan Pablo García Vázquez

Vo.Bo. de subdirector(es) de Unidad(es) Académica(s)

Humberto Cervantes De Ávila
Daniela Mercedes Martínez Platas
Noemí Hernández Hernández

Fecha: 23 de febrero de 2021

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Análisis de Tecnologías Emergentes sitúa el estudio de una variedad de herramientas actuales que facilitan la comprensión y solución de problemas complejos tanto en la industria como en la sociedad a través del uso de las tecnologías de software y hardware. Esto le permitirá al estudiante que conozca, analice e interactúe con la diversidad de tecnologías emergentes para identificar las mejores soluciones a problemáticas de usuarios finales. Esta asignatura es de carácter obligatorio de la etapa disciplinaria y contribuye al área de conocimiento Métodos y Tecnologías de Software.

III. COMPETENCIA GENERAL DE LA UNIDAD DE APRENDIZAJE

Valorar las tecnologías de cómputo emergentes a partir del análisis de sus características y aplicaciones, para reconocer los avances científicos y tecnológicos en el diseño de soluciones de cómputo innovadoras y efectivas, con una actitud crítica y objetiva.

IV. EVIDENCIA(S) DE APRENDIZAJE

Un reporte escrito descriptivo donde se proponga una solución tecnológica a partir del análisis del contexto de un usuario.

V. DESARROLLO POR UNIDADES
UNIDAD I. Introducción a las tecnologías emergentes

Competencia:

Analizar la interacción y dependencia de los elementos que incluyen las tecnologías emergentes a partir de sus componentes base para reconocer su funcionamiento, capacidades y aportaciones a las soluciones de usuarios, con una actitud reflexiva.

Contenido:

- 1.1. Conceptualización y contextualización de tecnologías emergentes
- 1.2. Componentes base de tecnologías emergentes
 - 1.2.1. Sensores
 - 1.2.2. Datos
 - 1.2.3. Redes y protocolos de comunicación
 - 1.2.4. Dispositivos de cómputo y embebidos para sistemas inteligentes

Duración: 12 horas

UNIDAD II. Tecnologías Emergentes

Competencia:

Analizar las tecnologías emergentes a partir de sus características y aplicaciones para reconocer su potencial en la atención de problemas y necesidades de usuarios con soluciones de cómputo consolidadas y efectivas, con pensamiento crítico y propositivo.

Contenido:

- 2.1. Aprendizaje automático
- 2.2. Datos masivos
- 2.3. Ambientes inteligentes
- 2.4. Cómputo móvil y vestible
- 2.5. Cómputo en la nube
- 2.6. Internet de las cosas
- 2.7. Realidad Virtual
- 2.8. Realidad aumentada
- 2.9. Visión computacional

Duración: 20 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

| No. | Nombre de la Práctica | Procedimiento | Recursos de Apoyo | Duración |
|-----------------|---|---|--|----------|
| UNIDAD I | | | | |
| 1 | Sensor y datos (Arduino con internet) | <ol style="list-style-type: none"> 1. Caracterizar el arduino, sensores y actuadores. 2. Instalar el software de arduino en una pc. 3. Prueba de funcionalidad del arduino. 4. Crear el primer programa para mandar datos al puerto. 5. Conexión de un sensor al arduino. 6. Modificar el programa para leer datos del puerto del arduino. 7. Comprobar que la lectura sea correcta. | <ul style="list-style-type: none"> • Arduino, sensores y actuadores (p. ej. sensor de temperatura, humedad, foto resistencia, etc. • Juego de resistencias y leds. • Documento de práctica. • Computadora. • Internet inalámbrico. | 4 horas |
| 2 | Redes (aplicación cliente-servidor) | <ol style="list-style-type: none"> 1. Crear una cuenta en su servicio de nube. 2. Crear y configurar los Apikeys. 3. Inicialice los dispositivos. 4. Establezca las jerarquías. 5. Cargar el programa que leer uno o más sensores. 6. Compruebe la conectividad. 7. Verificar que el arduino maestro esté recibiendo los datos de su sensor. | <ul style="list-style-type: none"> • Internet inalámbrico. • Arduino, sensores y actuadores (p. ej. sensor de temperatura, humedad, foto resistencia, etc.). • Computadora. • Documento de práctica • Cuenta en servicio de nube (p. ej. Amazon, Azure, Google Cloud, Carriots) | 4 horas |
| 3 | Demostración de internet de las cosas con almacenaje en la nube. | <ol style="list-style-type: none"> 1. Crear un nuevo dispositivo en su servicio de nube. 2. Descargar y configurar su plataforma de colaboración para el desarrollo de APIs en su pc. | <ul style="list-style-type: none"> • Internet Inalámbrico. • Computadora. • Conjunto de arduinos, sensores y actuadores (p. ej. sensor de temperatura, humedad, foto resistencia, etc.) | 6 horas |

| | | | | |
|---|---|---|---|---------|
| | | <ol style="list-style-type: none"> 3. Verifique que los datos aparezcan en el stream. 4. Enviar datos de sensores desde el arduino. 5. Modificar la arquitectura del arduino para que envíe datos de los sensores al presionar un botón. 6. Verificar que los datos aparezcan en el stream. | <ul style="list-style-type: none"> • Cuenta en servicio de nube (p. ej. Amazon, Azure, Google Cloud, carriots) • Software (p. ej. carriots, postman, Zoho). | |
| 4 | Aprendizaje automático | <ol style="list-style-type: none"> 1. Instalación de librerías. Nota: la instalación estará en función de los algoritmos a utilizar (p. ej. e1071 para SVM, Shiny) 2. Crear código cliente (p. ej. dashboard) 3. Crear código servidor (uso de algoritmos de aprendizaje supervisado o no supervisado considerando las especificaciones del proyecto a realizar) 4. Crear conexión con el servicio de nube. 5. Realizar pruebas con dashboard. | <ul style="list-style-type: none"> • Internet • Computadora • Software (R, RStudio, Shiny o Orange) • Cuenta en un servicio de nube (p. ej. Amazon, Azure, Google Cloud) • Documento de Práctica | 4 horas |
| 5 | Datos masivos, ambientes inteligentes y cómputo móvil y vestible | <ol style="list-style-type: none"> 1. Cargar la aplicación proporcionada por el instructor. 2. Lectura de datos provenientes de sensores (p. ej. magnetómetro, acelerómetro, PPG, SP02) 3. Almacenamiento de los datos en la nube. 4. Lectura y ejecución del programa otorgado por el | <ul style="list-style-type: none"> • Dispositivo móvil o vestible (p. ej. smartphone, mindset para EEG,) • Cuenta con un servicio de nube (p. ej. Amazon, Azure, Google Cloud) • Computadora • Software (R, RStudio, Shiny u Orange) • Internet. | 6 horas |

| | | | | |
|---|-------------------------------------|---|--|---------|
| | | <p>profesor.</p> <ol style="list-style-type: none"> 5. Instalación de librerías. Nota: la instalación estará en función de los algoritmos a utilizar (p. ej. e1071 para SVM, Shiny) 6. Crear código cliente (p. ej. dashboard) 7. Crear código servidor (uso de algoritmos de aprendizaje supervisado o no supervisado considerando las especificaciones del proyecto a realizar) 8. Crear conexión con el servicio de nube. 9. Realizar el análisis de los datos. | <ul style="list-style-type: none"> • Documento de práctica. | |
| 6 | Realidad Virtual y Aumentada | <ol style="list-style-type: none"> 1. Analiza el caso de uso seleccionado para la práctica. 2. Codifica la solución al problema planteado implementando realidad virtual y aumentada. 3. Entrega el código y el reporte de la práctica al docente para su evaluación | <ul style="list-style-type: none"> • Internet • Software para desarrollo de realidad virtual o aumentada (ARtoolkit, JIRO) • Software para redactar el reporte • Dispositivo para ejecutar el programa desarrollado | 4 horas |
| 7 | Visión computacional | <ol style="list-style-type: none"> 1. Descargar el dataset indicado por el instructor. 2. Instalar en su computadora el entorno anaconda. 3. Instalar las librerías indicadas por el instructor. 4. Abrir y ejecutar el código otorgado por su instructor. 5. Descargue imágenes de los objetos que su instructor le indique de la Web. | <ul style="list-style-type: none"> • Software (anaconda, tensorflow, keras, python, opencv) • Dataset indicado por el maestro (fruit 360) • Código de la práctica • Computadora con cámara o cámara usb o jetson nano. • Documento de la práctica. • Internet. | 4 horas |

| | | | | |
|--|--|---|--|--|
| | | <p>6. Compile el programa y descargue el modelo generado.</p> <p>7. Ejecute el modelo y alimente con las imágenes que usted descargó.</p> <p>Compruebe su utilidad con la cámara adjunta a su computadora o una cámara externa.</p> | | |
|--|--|---|--|--|

VII. MÉTODO DE TRABAJO

Encuadre: El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

Estrategia de enseñanza (docente):

- Técnica expositiva
- Resolución de problemas
- Estudios de caso
- Instrucción guiada
- Uso de TIC

Estrategia de aprendizaje (alumno):

- Técnica expositiva
- Resolución de problemas
- Estudios de caso
- Investigación documental
- Prácticas de laboratorio
- Aprendizaje basado en proyecto
- Trabajo colaborativo
- Uso de TIC
- Organizadores gráficos (mapas conceptuales, cuadros comparativos-descriptivos, cuadros sinóptico, entre otros)

VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

Criterios de acreditación

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir con los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

Criterios de evaluación

- Evaluaciones parciales..... 30%
- Prácticas de laboratorio..... 40%
- Reporte integrador de tecnologías emergentes30%
- Total.....100%**

IX. REFERENCIAS

Básicas

- Diamandis, P. H. & Kotler, S. (2020). *The future is faster than you think: how converging technologies are transforming business, industries, and our lives*. London: Simon and Shuster.
- Garrell, A. y Guilera, L. (2019). *La industria 4.0 en la sociedad digital*. España: Marge books.
- Neves, B. B., y Vetere, F. (2019). *Ageing and digital technology*. Singapore: Springer Singapore.
- Osinski, I. C. y Costas, C. S. L. (2018). *Análisis de datos en investigación. Primeros pasos*. España: Universidad Miguel Hernández.
- Perdomo, C.J.V. (2016). *Análisis de datos*. México: CIDE.
- Tehranipoor, M., Forte, D., Rose, G. S. & Bhunia, S. (2017). *Security opportunities in nano devices and emerging technologies*. New York, United States: CRC Press.

Complementarias

- Cortina, A. y Serra, M. Á. (2016). *Humanidad infinita: Desafíos éticos de las tecnologías emergentes*. Irlanda: Ediciones Internacionales Universitarias.
- Martínez, C. I. P., Poveda, A. C., y Moreno, S. P. F. (2019). *Analysis of science, technology, and innovation in emerging economies*. Suiza: Springer International Publishing.
- Kurniawan, A. (2021). Arduino Nano 33 IoT Board Development. In *Beginning Arduino Nano 33 IoT* (pp. 23-78). Apress, Berkeley, CA.
- Ohri, A. (2014). *R for cloud computing: An approach for data scientists*. Springer.

X. PERFIL DEL DOCENTE

El docente que imparta la unidad de aprendizaje Análisis de Tecnologías Emergentes debe contar con título de Ingeniero en Computación o área afín, preferentemente con posgrado en Ciencias de la Computación o a fin, con dos años de experiencia en el área de tecnologías emergentes y docencia. Ser proactivo, analítico y que fomente el trabajo en equipo.