



Guía para el profesor

Fundamentos y Aplicaciones de IA
Clave MSCA2101



Índice

Información general del curso	1
Metodología	2
Evaluación.....	3
Bibliografía.....	4
Tips importantes.....	6
Temario	7
Notas de enseñanza	8
Fase I. Planteamiento.....	33
Fase II. Ejecución.....	35
Fase III. Resultados	37

Información general del curso

Modalidades

- Clave: MSCA2101
- Modalidad: Connect

Competencia del certificado

Aplica conocimientos de inteligencia artificial integrando sus principales herramientas para el desarrollo de aplicaciones para la industria.





Metodología

Este certificado Connect ha sido diseñado para ser facilitado por un docente líder con experiencia profesional en el ámbito laboral, capaz de compartir con los estudiantes no solo sus conocimientos, sino también sus vivencias y mejores prácticas. Tu papel como docente es clave para enriquecer la formación de los participantes, guiándolos activamente a través de las actividades y aportando una visión práctica y aplicada de los contenidos.

La experiencia Connect promueve la interacción virtual entre estudiantes de distintos campus de la Universidad Tecmilenio, lo que permite generar un intercambio valioso de perspectivas y contextos regionales. En cada sesión virtual sincrónica, tendrás la oportunidad de fungir como guía y facilitador del aprendizaje, propiciando la participación activa y el diálogo significativo.

El certificado tiene una duración semestral y se estructura en sesiones semanales de tres horas cada una, impartidas mediante una herramienta de videoconferencia. La asistencia de los estudiantes es fundamental para el aprovechamiento del curso; sin embargo, en caso de que no puedan asistir, contarán con el recurso de la grabación para ponerse al día.



Evaluación

A continuación, puedes revisar el detalle de la evaluación de este certificado:

Evaluable	Ponderación
Actividad 1	7
Actividad 2	7
Actividad 3	7
Fase I	12
Actividad 4	7
Actividad 5	7
Actividad 6	7
Fase II	12
Actividad 7	7
Actividad 8	7
Actividad 9	7
Fase III	13
Total	100



Bibliografía

Módulo 1

- Kaplan, J. (2016). *Inteligencia Artificial: Lo que todo el mundo debe saber*. España: Teell Editorial. ISBN: 9788416511600

Módulo 2

- Matthes, E. (2019). *Python Crash Course: A Hands-On, Project-Based Introduction to Programming* (2a ed.). Estados Unidos: No Starch Press. ISBN: 9781593279288.

Módulo 3

- Iansiti, M. (2020). *Competing In The Age of AI: Strategy and Leadership When Algorithms and Networks Run the World*. Estados Unidos: Harvard Business Review Press. ISBN: 9781633697621

Software

Módulo 1

- Plataforma de desarrollo académico de IBM: <https://skills-academy.comprehend.ibm.com/>

Módulo 2

- Plataforma para el desarrollo de ciencia de datos y aprendizaje automático en lenguaje Python y R, incluye Jupyter Notebook: <https://www.anaconda.com/products/individual>

Módulo 3

- Plataforma de servicios (PaaS) que integra los servicios de IBM Watson: <https://cloud.ibm.com/>
- Servicio de IBM Watson, motor de búsqueda cognitiva y análisis de contenido a las aplicaciones para identificar patrones y tendencias, empleando inteligencia artificial: <https://www.ibm.com/mx-es/cloud/watson-discovery>
- Servicio de IBM Watson, interfaz de lenguaje natural e inteligencia artificial para crear, entrenar e implementar interacciones conversacionales en cualquier aplicación, dispositivo o canal. Las aplicaciones incluyen agentes virtuales y bots de

conversación: <https://www.ibm.com/mx-es/cloud/watson-assistant>

- Servicio y herramienta de IBM, reconocimiento automático de voz a texto utilizando algoritmos de inteligencia artificial: <https://www.ibm.com/mx-es/cloud/watson-speech-to-text>
- Servicio de IBM Watson dedicado a la creación de clasificadores predeterminados o personalizados y desarrollo de aplicaciones inteligentes que analizan el contenido visual de imágenes o fotogramas para comprender lo que sucede en una escena: <https://www.ibm.com/mx-es/cloud/watson-visual-recognition/pricing>





Tips importantes

Durante esta experiencia de aprendizaje, los estudiantes tendrán la oportunidad de familiarizarse con la plataforma de infraestructura en la nube de IBM (IBM Cloud) como parte del desarrollo de soluciones de inteligencia artificial en la actividad correspondiente a la semana 3. Asimismo, en la semana 9, podrán obtener una insignia digital al aprobar el curso *Getting Started with Artificial Intelligence*. Como docente, es importante que facilite el acceso a estos recursos, motive su exploración y brinde acompañamiento en la integración de estos elementos al aprendizaje del curso.



Temario

Tema	Título
Tema 1	La inteligencia artificial en el mundo actual
Tema 2	Historia y evolución de la inteligencia artificial
Tema 3	Definición de inteligencia artificial
Tema 4	La inteligencia artificial y la ciencia de datos
Tema 5	Inteligencia artificial simbólica
Tema 6	Introducción al aprendizaje automático
Tema 7	Tecnologías avanzadas de la inteligencia artificial
Tema 8	Desafíos y oportunidades para la inteligencia artificial
Tema 9	Conceptos generales y lenguajes de programación para inteligencia artificial
Tema 10	Introducción al lenguaje de programación multiplataforma en Python
Tema 11	Estructuras de datos en el lenguaje de programación multiplataforma
Tema 12	Sentencias condicionales y de control de flujo (bucles) en Python
Tema 13	Funciones en Python
Tema 14	Manejo de archivos, objetos y clases en Python
Tema 15	Librerías para el manejo de datos en Python
Tema 16	Librerías para la visualización de los datos en Python
Tema 17	Aplicaciones básicas de la inteligencia artificial
Tema 18	Soluciones y tecnologías para hacer inteligencia artificial
Tema 19	Aplicaciones conversacionales básicas
Tema 20	Uso de aplicaciones conversacionales
Tema 21	Campos de aplicación de la inteligencia artificial
Tema 22	Clasificadores básicos de imágenes
Tema 23	Uso de los clasificadores de imágenes
Tema 24	Implementación de una aplicación de inteligencia artificial en la web



Notas de enseñanza

Las notas de enseñanza son estrategias didácticas sugeridas que tienen como propósito apoyar al docente en la mejora de la experiencia de aprendizaje al impartir el curso. Estas recomendaciones ofrecen ideas prácticas para facilitar la comprensión de los contenidos, fomentar la participación activa del estudiante y fortalecer el vínculo entre la teoría y su aplicación en contextos reales.

Tema 1 La inteligencia artificial en el mundo actual

Enfoque para abordar la clase:

- **Comienza con una reflexión:** Plantea preguntas abiertas como “¿Qué entendemos por inteligencia artificial hoy?” y “¿En qué áreas ya interactúan con IA sin notarlo?” para romper mitos y generar conexión con su experiencia diaria.
- **Clasificación funcional:** Expón los tipos de IA (específica, extendida y general) mediante ejemplos concretos. Usa analogías cotidianas para diferenciar su alcance y limitaciones.
- **Conexión con la vida real:** Resalta que la IA no es ciencia ficción, sino una herramienta en uso hoy en día en salud, educación, industria, banca, entretenimiento, etc.

Ejemplos recomendados:

- **IA específica:** asistentes virtuales (Siri, Alexa), recomendaciones de contenido (Netflix, Amazon).
- **IA extendida:** vehículos autónomos, análisis financiero, minería de datos en yacimientos.
- **IA general (hipotética):** tutor personalizado inteligente o sistemas con autonomía de decisión al nivel humano.

Actividades sugeridas:

1. **Lluvia de ideas:** Pide al grupo que enumere aplicaciones actuales de la IA que conozcan. Después, clasifíquenlas según el tipo de IA.
2. **Estudio de caso:** Presenta un caso (por ejemplo, Watson en medicina) y analicen cómo aumentó la capacidad humana y qué retos éticos surgieron.
3. **Mini debate:** ¿La IA genera más beneficios o riesgos para la sociedad? Organiza un breve debate argumentado con base en los subcampos estudiados (PLN, visión artificial, robótica, etc.).
4. **Mapa de aplicaciones:** En equipos, elaboren un mapa visual que conecte los subcampos de la IA con sectores específicos de la industria.

Tema 2 Historia y evolución de la inteligencia artificial

Enfoque para abordar la clase:

- **Narrativo histórico:** Presenta la historia de la IA como una narrativa, resaltando sus momentos clave, personajes influyentes (McCarthy, Minsky, Simon, Hinton, etc.) y contextos sociotécnicos como la guerra, los avances en hardware y los cambios económicos.
- **Contraste de paradigmas:** Subraya la diferencia entre los sistemas simbólicos y el aprendizaje automático, utilizando ejemplos accesibles (e.g., SHRDLU vs. redes neuronales modernas).
- **Énfasis en los ciclos:** Explica los "inviernos de la IA" como ciclos naturales de la ciencia aplicada, conectándolos con las expectativas tecnológicas no cumplidas y los factores económicos.

Ejemplos recomendados:

- **SHRDLU y el uso de lógica simbólica.**
- **El perceptrón de Rosenblatt y su cobertura mediática.**
- **Deep Blue vs. Kasparov** como hito en el razonamiento computacional.
- **Watson en Jeopardy** como demostración del PLN.
- **Vehículos autónomos y los DARPA Grand Challenges** para ilustrar desafíos técnicos reales.

Actividades sugeridas:

1. **Línea del tiempo colaborativa:** En grupos, los estudiantes colocan en una línea del tiempo los eventos clave de la historia de la IA con breves descripciones.
2. **Debate guiado:** ¿Sistemas simbólicos o aprendizaje automático? ¿Cuál paradigma ha sido más transformador? Invita a los alumnos a defender posturas en función de casos históricos.
3. **Análisis de mitos:** Presenta titulares exagerados de épocas pasadas (como el NYT sobre el perceptrón) y analiza con el grupo las expectativas frente a las capacidades reales.
4. **Reflexión final:** Cierra con una pregunta abierta: "¿Qué hito futuro podría marcar la próxima gran etapa de la IA?"

Tema 3 Definición de Inteligencia Artificial**Enfoque para abordar la clase:**

- **Inicio reflexivo:** Abre la sesión preguntando "¿Qué es para ti la inteligencia?" y "¿Una máquina puede ser inteligente?". Esto permite identificar ideas previas y contrastarlas con definiciones más formales.
- **Comparación gradual:** Introduce las definiciones de IA progresivamente, contrastando sus elementos clave y relacionándolos con conceptos como razonamiento, generalización y aprendizaje.
- **Enlace con la cognición humana:** Usa la teoría de inteligencias múltiples de Gardner como puente para explorar las limitaciones de las definiciones tradicionales de IA.

Ejemplos recomendados:

- **Ejemplo de IA simbólica:** Un sistema experto que evalúa si un usuario está inscrito en una base de datos.
- **Ejemplo de aprendizaje automático:** Un sistema que recomienda películas en una plataforma de streaming.
- **Ejemplo híbrido:** El programa Dr. Fill para resolver crucigramas y el sistema financiero de Taiger.

Actividades sugeridas:

1. **Mapa conceptual colaborativo:** Elabora un mapa con los conceptos clave de IA: definición, tipos (específica, extendida, general), paradigmas (simbólico, conexionista), y sus relaciones.
2. **Taller de clasificación:** Da ejemplos reales de IA y pide que los estudiantes identifiquen si aplican una IA específica, extendida o general, y bajo qué paradigma funcionan.
3. **Comparación de paradigmas:** En equipos, asigna un caso a resolver (como el robot futbolista) y pideles que propongan cómo abordarlo desde ambos enfoques: simbólico y conexionista.
4. **Debate corto:** ¿Puede la inteligencia humana modelarse completamente en una máquina? Promueve argumentos basados en lo discutido.

Tema 4 La inteligencia artificial y ciencia de datos

Enfoque para abordar la clase:

- **Inicio con ejemplos impactantes:** Muestra casos como Netflix o Starbucks para evidenciar cómo las decisiones guiadas por datos pueden transformar negocios.
- **Enlace conceptual:** Aclara cómo la IA necesita datos para "aprender" y por qué la calidad y estructura de los datos es crítica para su éxito.
- **Diferenciación gradual:** Explica los tipos de datos (estructurados, no estructurados, etc.) y herramientas (bases de datos, lagos, almacenes) con ejemplos reales.

Ejemplos recomendados:

- **Starbucks y el uso de SIG** para elegir ubicaciones.
- **Netflix y la creación de "House of Cards"** mediante Big Data.
- **Hadoop y bases NoSQL (MongoDB)** como herramientas clave para grandes volúmenes de datos.

Actividades sugeridas:

1. **Clasificación de datos:** Presenta distintos tipos de contenido (imágenes, tablas, registros XML, series temporales) y pide a los estudiantes clasificarlos según la tipología de datos.
2. **Simulación de CRISP-DM:** Organiza al grupo en equipos y asigna un problema sencillo (por ejemplo, optimizar un menú de cafetería). Que sigan las fases de la metodología para definir cómo usarían los datos.

3. **Mapa de herramientas:** Elaboren un diagrama comparativo entre bases de datos relacionales, NoSQL, almacenes y lagos de datos.
4. **Estudio de caso:** Analicen el caso de Netflix, identificando qué tipos de datos utilizaron, cómo los analizaron y qué decisiones empresariales tomaron.

Tema 5 Inteligencia artificial simbólica

Enfoque para abordar la clase:

- **Desmitifica la “obsolescencia” simbólica:** Comienza aclarando que la IA simbólica no está relegada al pasado, sino que sigue siendo útil y se combina con el enfoque conexionista en aplicaciones concretas.
- **Ejemplificación técnica:** Explica la lógica detrás de la planificación automática con ejemplos simples (organizar tareas diarias, ensamblar piezas, rutas de entrega) antes de pasar a modelos complejos como la planificación espacial.
- **Transversalidad del aprendizaje de ontologías:** Conecta su uso con campos diversos (salud, gobierno, e-commerce) y destaca cómo actúa como puente entre texto natural y conocimiento computacional.

Ejemplos recomendados:

- **Planificación automática:** Construcción naval, ensamblaje de aviones, ejecución general de juegos.
- **Aprendizaje de ontologías:** Web semántica en Google o Yahoo, sistemas de gobierno interconectados, diagnósticos médicos asistidos.
- **Sistema híbrido:** Ontologías combinadas con aprendizaje automático para crear motores de búsqueda semánticos o sistemas de recomendación.

Actividades sugeridas:

1. **Estudio de caso en equipo:** Plantea un problema como la programación de una fábrica o la creación de un buscador semántico. Pide que los estudiantes propongan cómo la IA simbólica y/o híbrida lo abordaría.
2. **Diagrama de flujo simbólico:** Elaboren un esquema de planificación automática para una tarea cotidiana, marcando el estado inicial, operaciones posibles, objetivo y heurísticas.
3. **Análisis de noticias:** Selecciona una noticia tecnológica actual que involucre razonamiento simbólico (por ejemplo, regulación automatizada o sistemas legales inteligentes) y discútanla en clase.
4. **Debate guiado:** ¿Debe enfocarse el desarrollo de IA solo en paradigmas conexionistas o es necesario fortalecer los híbridos?

Tema 6 Introducción al aprendizaje automático

Cómo abordar la clase:

- Inicia con una breve narrativa sobre el caso de Coca-Cola y cómo mejoró el registro de usuarios usando IA para leer códigos en tapas. Usa esto para introducir la necesidad de que las máquinas “aprendan” a partir de datos.
- Recalca que el aprendizaje automático es una rama de la inteligencia artificial que ha revolucionado la forma en que se procesan decisiones y se automatizan tareas.
- Evita explicar primero los modelos técnicos. En su lugar, usa ejemplos y problemas de la vida real que los estudiantes puedan identificar como problemas que requieren aprendizaje automático.

Puntos clave a explicar:

- 1. Definición y propósito del aprendizaje automático.**
 - Diferencia entre programación tradicional y modelos que aprenden de datos.
 - Ejemplos cotidianos: recomendaciones de contenido, reconocimiento de voz, filtros de spam.
- 2. Relación con estadística y probabilidad.**
 - Distribución normal y desviación estándar.
 - Teorema de Bayes y su aplicación en inferencias.
 - Correlación de Pearson como base para identificar patrones.
- 3. Proceso de aprendizaje automático.**
 - Desde el entendimiento del problema hasta la implementación del modelo.
 - Destacar fases: preparación de datos, entrenamiento, evaluación y despliegue.
 - Subrayar la importancia del análisis de errores para mejorar modelos.
- 4. Tipos de aprendizaje:**
 - Supervisado: requiere datos etiquetados.
 - No supervisado: descubre patrones sin etiquetas.
 - Por refuerzo: aprendizaje basado en recompensas.
 - Semi-supervisado: combinación de los anteriores.
- 5. Redes neuronales y aprendizaje profundo.**
 - Diferencias entre aprendizaje automático y profundo.
 - Presentar el perceptrón como origen histórico.
 - Explicar de forma visual las capas de las redes neuronales.

Preguntas para fomentar la discusión:

- ¿En qué casos de su vida diaria creen que ya se usa aprendizaje automático?

- ¿Por qué sería preferible entrenar un modelo en vez de programar manualmente todas las reglas?
- ¿Cuáles serían los riesgos de usar un modelo de ML sin comprender los datos?

Actividades sugeridas:

1. **Ejercicio práctico con datos simples.**
 - Simulación de clasificación con datos ficticios de frutas: color, tamaño, forma.
 - Visualización de la relación entre variables (correlación).
2. **Análisis de caso real.**
 - Presentar el caso de Coca-Cola y discutir en grupo qué tipo de aprendizaje automático se utilizó.
 - Buscar y compartir un ejemplo adicional de otra empresa que use ML.
3. **Reflexión guiada.**
 - ¿Qué retos enfrentarías al implementar un sistema que necesita aprender?
 - ¿Cómo sabrías si un modelo es "bueno"?

Tema 7 Tecnologías avanzadas de inteligencia artificial

Recomendaciones para abordar la clase

- **Contextualización inicial:** Comience la sesión preguntando a los estudiantes sobre aplicaciones cotidianas que utilizan inteligencia artificial, como recomendaciones en plataformas de streaming o asistentes virtuales. Esto ayudará a conectar el tema con su experiencia diaria.
- **Definición clara:** Explique que el aprendizaje automático es una rama de la inteligencia artificial que permite a las máquinas aprender de los datos y mejorar su rendimiento con el tiempo sin ser programadas explícitamente para cada tarea.
- **Diferenciación de conceptos:** Aclare la diferencia entre inteligencia artificial y aprendizaje automático, enfatizando que este último es un subconjunto de la IA centrado en el aprendizaje a partir de datos.
- **Tipos de aprendizaje:** Presente los principales tipos de aprendizaje automático: supervisado, no supervisado y por refuerzo. Utilice ejemplos sencillos para ilustrar cada uno, como la clasificación de correos electrónicos (supervisado), agrupación de clientes por comportamiento de compra (no supervisado) y entrenamiento de agentes en videojuegos (por refuerzo).
- **Aplicaciones prácticas:** Muestre casos de uso reales del aprendizaje automático en diversas industrias, como la detección de fraudes en finanzas, diagnósticos médicos asistidos por IA y sistemas de recomendación en comercio electrónico.

Sugerencias de ejemplos y actividades

- **Análisis de casos:** Presente estudios de caso donde el aprendizaje automático haya tenido un impacto significativo, como el uso de redes neuronales en el diagnóstico de enfermedades o en vehículos autónomos.
- **Actividad práctica:** Organice una actividad donde los estudiantes clasifiquen manualmente un conjunto de datos sencillo (por ejemplo, tipos de frutas por características) y luego comparen sus resultados con los obtenidos por un modelo básico de aprendizaje automático.
- **Discusión ética:** Fomente un debate sobre las implicaciones éticas del uso del aprendizaje automático, abordando temas como la privacidad de los datos y los sesgos en los algoritmos.

Cierre de la sesión

- **Resumen:** Repase los conceptos clave abordados durante la clase, reforzando la comprensión de los estudiantes.
- **Reflexión final:** Invite a los estudiantes a reflexionar sobre cómo el aprendizaje automático podría influir en sus futuras profesiones y en la sociedad en general.
- **Recomendaciones adicionales:** Sugiera recursos para profundizar en el tema, como cursos en línea, artículos académicos o tutoriales prácticos.

Tema 8 Desafíos y oportunidades para la inteligencia artificial

Enfoque Pedagógico

Este tema invita a una reflexión crítica sobre el impacto de la inteligencia artificial (IA) en la sociedad. Se recomienda fomentar un ambiente de diálogo donde los estudiantes puedan analizar tanto los beneficios como los riesgos asociados al uso de la IA, considerando aspectos éticos, sociales y tecnológicos.

Recomendaciones para Abordar la Clase

1. **Iniciar con un Caso Real Impactante**
Comience la sesión presentando el caso de Jacky Alciné, quien en 2015 descubrió que Google Photos etiquetó erróneamente fotos suyas y de una amiga como "gorilas", evidenciando sesgos raciales en los algoritmos de reconocimiento facial. Este ejemplo servirá para ilustrar cómo los sesgos en los datos pueden tener consecuencias graves y reales.
2. **Explorar Aplicaciones Positivas de la IA**
Destaque proyectos donde la IA ha sido utilizada para el bien social, como:
 - **Rainforest Connection:** Utiliza dispositivos acústicos alimentados por energía solar para detectar y prevenir la tala ilegal en selvas tropicales.
 - **Brain Power y Afectiva:** Desarrollo de gafas inteligentes que ayudan a niños y adultos con autismo a reconocer y responder a emociones en tiempo real.
3. **Debatir sobre Desafíos Éticos y Sociales**
Promueva discusiones sobre temas como:
 - Privacidad y vigilancia en la era digital.
 - Manipulación del comportamiento a través de algoritmos.

- Opacidad y sesgos en sistemas de IA.
- Impacto de la automatización en el empleo.
- Responsabilidad en el uso de sistemas autónomos.

Actividades Sugeridas

- **Análisis de Casos:** Divida a los estudiantes en grupos y asígneles diferentes casos para analizar, considerando los aspectos éticos, sociales y tecnológicos involucrados.
- **Debate Estructurado:** Organice un debate donde un grupo defienda las oportunidades que ofrece la IA y otro exponga los desafíos y riesgos asociados.
- **Simulación de Políticas Públicas:** Pida a los estudiantes que diseñen propuestas de políticas para regular el uso ético de la IA en diferentes sectores

Cierre de la Sesión

Concluya enfatizando la importancia de una IA desarrollada y utilizada de manera ética y responsable. Subraye que, si bien la IA ofrece oportunidades significativas para mejorar diversos aspectos de la sociedad, es crucial abordar proactivamente los desafíos que presenta para garantizar que su impacto sea positivo y equitativo.

Tema 9 Conceptos generales y lenguajes de programación para inteligencia artificial

Enfoque para abordar la clase

Aborda esta clase como un recorrido histórico y conceptual que conecte los orígenes de la computación con las tecnologías de inteligencia artificial actuales. Inicia con una narrativa que destaque los aportes de figuras clave como Ada Lovelace, Babbage y Turing, para después introducir la evolución de los lenguajes de programación y su relación con la IA. Resalta la conexión entre el pensamiento lógico-matemático y los lenguajes como Python, Lisp, Prolog y R. Fomenta la reflexión crítica sobre cómo los avances técnicos están relacionados con necesidades sociales y tecnológicas específicas. Es importante mantener un equilibrio entre teoría, historia y aplicaciones prácticas contemporáneas.

Ejemplos recomendados

- Comparar un algoritmo cotidiano (como una receta de cocina) con un algoritmo de búsqueda computacional para introducir el concepto de algoritmos.
- Mostrar fragmentos de código simples en Python, Lisp y Prolog para ilustrar diferentes paradigmas (imperativo, funcional, lógico).
- Utilizar un ejemplo práctico como el reconocimiento facial para explicar cómo interactúan lenguajes y marcos de trabajo (Python + TensorFlow).
- Mencionar cómo herramientas como Pandas o Matplotlib ayudan a resolver problemas concretos en IA, como el análisis de datos de ventas o predicción de tendencias.

Actividades sugeridas

- **Línea del tiempo interactiva:** Pida al alumnado que, en equipos, construya una línea del tiempo sobre los hitos históricos más importantes mencionados (de Jacquard a la computación cuántica).
- **Análisis de código:** Presentar pequeños fragmentos de código en diferentes lenguajes (Python, R, Prolog) y pedirles que identifiquen su propósito y estructura.
- **Debate breve:** "¿Lenguajes específicos para IA o lenguajes de propósito general con librerías especializadas? ¿Qué conviene más y por qué?"

- **Mini proyecto grupal:** Diseñar una propuesta de solución basada en IA para un problema cotidiano (recomendador de películas, chatbot, etc.), investigando qué lenguajes y herramientas usarían.

Tema 10 Introducción al lenguaje de programación multiplataforma en Python

Enfoque para abordar la clase

Plantea la sesión como una experiencia práctica desde el primer momento, donde el estudiante vea resultados inmediatos al escribir sus primeras líneas de código en Python. Refuerza la idea de que Python es un lenguaje accesible, con una sintaxis limpia y altamente legible. Resalta su utilidad en múltiples campos (desde ciencia de datos hasta automatización y desarrollo web), así como su presencia en aplicaciones reales y populares. Es importante transmitir que aprender Python no solo es posible para cualquiera, sino también útil y relevante. Invita a los estudiantes a reflexionar sobre su experiencia con la instalación y uso de herramientas como Anaconda y Jupyter, promoviendo la autonomía técnica desde etapas tempranas.

Ejemplos recomendados

- El clásico "Hola mundo" como primer contacto con la sintaxis de Python.
- Uso de variables con diferentes tipos de datos: int, float, string y boolean.
- Mostrar cómo imprimir varias líneas con triple comillas y cómo calcular el área de un triángulo con operadores matemáticos básicos.
- Desplegar el "Zen de Python" para discutir los principios de simplicidad y legibilidad del lenguaje.
- Contrastar la salida de operaciones con / y // para ilustrar diferencias entre división decimal y truncada.

Actividades sugeridas

- **Exploración guiada en Jupyter Notebook:** Realiza una instalación en vivo de Anaconda y abre un cuaderno en Jupyter. Cada estudiante debe ejecutar el "Hola mundo" y modificarlo con su nombre o una frase propia.
- **Mini taller de tipos de datos:** Propón que cada estudiante declare una variable de cada tipo (entero, flotante, cadena, booleano) e imprima su valor y tipo usando type().
- **Actividad creativa con cadenas:** Pedir a los estudiantes que escriban un poema o diálogo utilizando ""texto"" y lo impriman con print().
- **Reto práctico:** Que los alumnos creen un pequeño programa que reciba dos números como base y altura, calcule el área de un triángulo y muestre ambos resultados: uno con / y otro con //.

Tema 11 Estructura de datos en el lenguaje de programación multiplataforma

Enfoque para abordar la clase

Enfoca la clase como un taller práctico donde el estudiante descubra que **las estructuras de datos son herramientas clave para organizar, procesar y acceder eficientemente a la información**. Destaca cómo cada estructura tiene usos específicos y ventajas particulares. Refuerza la importancia de dominar las estructuras básicas (listas, tuplas, diccionarios, conjuntos) como cimiento para resolver problemas reales. Apóyate en analogías cotidianas (como el estacionamiento, la agenda telefónica, el refrigerador, el túnel) para explicar el comportamiento lógico detrás de cada estructura. Incentiva el

uso de Jupyter Notebook para experimentar con el código en tiempo real y hacer más accesible la interacción con los datos.

Ejemplos recomendados

- Comparar una **lista de estudiantes** con una **tupla de dimensiones de una figura** para ilustrar mutabilidad vs inmutabilidad.
- Utilizar un **diccionario de estacionamiento** para introducir la lógica clave-valor.
- Mostrar el uso de **índices negativos y rebanadas** en cadenas para reforzar el manejo de secuencias.
- Ejemplificar con conjuntos (set) la eliminación de duplicados en una lista de palabras repetidas.
- Simular una **pila** con operaciones `append()` y `pop()` y una **cola** con `pop(o)` para representar el flujo de datos.

Actividades sugeridas

- **Exploración con listas:** Crea una lista vacía y pide al alumnado agregar, insertar y eliminar elementos usando métodos como `.append()`, `.insert()` y `.remove()`.
- **Rompecabezas de cadenas:** Proporciona una cadena desordenada y solicita al grupo usar rebanadas y métodos (`.upper()`, `.strip()`, `.replace()`) para “limpiar” y formatear el texto.
- **Reto de diccionarios:** En parejas, crear un diccionario que represente una agenda escolar (clave: materia, valor: horario) y que puedan consultar, actualizar y eliminar elementos.
- **Mini simulador de pila/cola:** Dividir la clase en equipos, asignar roles de “push/pop” y representar físicamente cómo funciona una pila o una cola. Luego, implementar esa lógica en código.
- **Visualización rápida:** Usar `print()` y `type()` para mostrar en tiempo real el tipo de estructura y sus elementos tras cada operación.

Tema 12 Sentencias condicionales y de control de flujo (bucles) en Python

Enfoque para abordar la clase

Aborda la clase como una oportunidad para **mostrar cómo un programa puede “tomar decisiones” y repetir acciones de forma automatizada**. Resalta que las estructuras condicionales (`if`, `if-else`, `if-elif-else`) permiten que los programas reaccionen de distintas maneras según los datos recibidos, y que los bucles (`for`, `while`) son herramientas para manejar tareas repetitivas. Enfatiza la importancia de la **indentación en Python como parte de la sintaxis obligatoria**. Usa ejemplos cotidianos (edad para votar, elegir ingredientes, repetir un saludo) para dar contexto al uso de cada estructura. Propicia una

clase activa con Jupyter Notebook, donde los estudiantes escriban y ejecuten cada ejemplo en tiempo real.

Ejemplos recomendados

- Comparar variables usando `==` y `!=`, mostrando diferencias entre asignación y comparación.
- Verificación de edad para votar usando `if-else`.
- Tarifa diferenciada según edad con `if-elif-else`.
- Uso de `and` y `or` con condiciones compuestas.
- Bucle `while` que solicita al usuario una entrada hasta que escriba "salir".
- Recorrido de listas con `for`, primero con `while` y luego con `for` para comparar eficiencia.
- Iteración sobre cadenas y diccionarios usando `.keys()`, `.values()` y `.items()`.
- Uso de `break`, `continue` y `else` en ciclos `for` con el ejemplo del queso.

Actividades sugeridas

- **Mini encuesta interactiva:** Usar `input()` y estructuras `if` para preguntar la edad del estudiante y dar una respuesta personalizada.
- **Clasificador por edades:** Pedir a cada alumno que codifique un programa que indique el costo de entrada a un evento según la edad, usando `if-elif-else`.
- **Eco interactivo:** Programar un bucle `while` donde el usuario escribe frases que se repiten hasta ingresar la palabra "salir".
- **Descomponer palabra:** Dar una palabra y pedir a los estudiantes que escriban un `for` para imprimir letra por letra.
- **Diccionario de usuario:** Crear un diccionario con datos de un perfil y mostrar claves, valores y pares con tres tipos de `for`.
- **Reto con listas:** Usar `for` para recorrer una lista de números y aplicar `continue` si el número es impar y `break` si se encuentra un cero.

Tema 13 Funciones en Python

Enfoque para abordar la clase

Aborda la sesión enfatizando que **las funciones permiten organizar y reutilizar el código**, haciendo que los programas sean más claros, eficientes y escalables. Refuerza la idea de que, al encapsular instrucciones dentro de funciones, los estudiantes adquieren buenas prácticas desde etapas tempranas. Presenta ejemplos simples primero, y avanza gradualmente hacia funciones con parámetros, valores por defecto y argumentos arbitrarios. Muestra cómo se pueden agrupar funciones

en módulos, destacando el valor del trabajo colaborativo y la reutilización. Usa Jupyter Notebook para favorecer una experiencia práctica inmediata y comprensible.

Ejemplos recomendados

- **Función básica:** `def saludar(): print("Hola")` seguida de su invocación.
- **Funciones con parámetros:** `def saludar(nombre)` y ejemplos con argumentos posicionales y por palabra clave.
- **Valores por defecto:** Funciones con parámetros opcionales (`nombre="invitado"`).
- **Valores de retorno:** `return` con operaciones como suma o concatenación.
- **Manipulación de listas:** Pasar listas como argumento y modificarlas dentro de funciones.
- **Tuplas arbitrarias:** `*args` para aceptar cualquier número de argumentos.
- **Funciones lambda:** Mostrar su utilidad para operaciones rápidas como transformaciones en listas.
- **Módulos:** Crear e importar un módulo simple con una o más funciones.

Actividades sugeridas

- **Mini taller de funciones personales:** Pide a los estudiantes que creen una función que salude, una que calcule algo (como edad o promedio) y otra que devuelva un mensaje con `return`.
- **Desafío con listas:** Crear una función que reciba una lista de nombres y devuelva una nueva con todos en mayúsculas.
- **Reto de hamburguesas:** Definir una función con `*args` que imprima una orden personalizada, al estilo del ejemplo de las hamburguesas.
- **Explora y reutiliza:** Proponer que agrupen varias funciones en un archivo `.py` y practiquen cómo importarlas en otro cuaderno.
- **Función lambda express:** Pedirles que definan una función lambda para sumar 3 números, o transformar cadenas de texto.

Tema 14 Manejo de archivos, objetos y clases en Python

Enfoque para abordar la clase

Comienza diferenciando claramente entre **almacenamiento en RAM** y **almacenamiento en disco**, destacando por qué es esencial trabajar con archivos en programas útiles y persistentes. Luego introduce el manejo de archivos de texto usando `open()`, `read()`, `write()` y `with`, explicando con ejemplos cómo escribir, leer y proteger archivos. Una vez dominado esto, transita hacia formatos estructurados como **CSV** y **JSON**, conectando su utilidad con bases de datos y aplicaciones web. Finalmente, introduce las clases y objetos con analogías concretas (como perros, robots o personajes de videojuego), remarcando cómo la **programación orientada a objetos permite modelar el mundo**

real. Mantén siempre una perspectiva práctica: cada nueva herramienta debe responder a un "¿para qué me sirve esto?".

Ejemplos recomendados

- **Manejo de archivos:**
 - Crear un archivo .txt con write() y print().
 - Leerlo con read(), readline(), readlines() y bucles for.
 - Escribir por fragmentos (chunk) para archivos largos.
 - Uso de try/except para evitar sobrescribir archivos.
 - Lectura/escritura de archivos .csv con el módulo csv.
 - Uso de json.dumps() y json.loads() con estructuras anidadas.
- **Programación orientada a objetos:**
 - Definición de la clase Perro con __init__, atributos y métodos (sentarse(), darse_la_vuelta()).
 - Creación de una instancia y acceso a sus atributos/métodos.
 - Mostrar cómo cada objeto puede tener valores distintos, pero el mismo comportamiento.

Actividades sugeridas

- **Archivo de texto interactivo:** Pide a los estudiantes que creen un programa que reciba un poema, lo guarde en un archivo y después lo lea línea por línea.
- **Exploración de CSV:** Crear un pequeño archivo de contactos o productos y escribir un lector de CSV que imprima cada línea como lista o diccionario.
- **Estructura JSON personalizada:** Crear un menú o perfil de usuario en un diccionario, guardarlo como JSON, leerlo de vuelta e imprimir sus partes clave.
- **Clase personalizada:** Proponer que cada estudiante cree una clase para modelar algo de su vida diaria (un celular, una mascota, un videojuego) e implemente al menos dos atributos y dos métodos.
- **Desafío de métodos:** Crear una clase Libro con atributos como titulo y autor, y métodos como mostrar_info() y marcar_leido().

Tema 15 Librerías para el Manejo de Datos en Python

Enfoque para abordar la clase

Comienza la clase destacando la historia de la **primera imagen de un agujero negro** como una forma de motivar a los estudiantes, mostrando cómo **Python y sus librerías** jugaron un papel esencial en un

hito científico global. Luego introduce **Numpy** como la herramienta base para trabajar con datos numéricos de forma eficiente, enfocándose en la estructura ndarray y su capacidad para operaciones vectorizadas. Posteriormente, presenta **Pandas** como una extensión poderosa orientada a datos heterogéneos y tabulares, ideal para análisis exploratorio, limpieza y transformación de datos. Haz énfasis en que **la combinación de Numpy + Pandas** es el estándar en ciencia de datos moderna. Utiliza ejemplos visuales y prácticos para mantener la atención y facilitar la comprensión.

Ejemplos recomendados

- **Numpy:**
 - Crear arreglos con `array()`, `arange()`, `ones()`, `full()`.
 - Consultar dimensiones y tipos (`.shape`, `.dtype`).
 - Uso de operaciones vectorizadas (suma, raíz cuadrada con `np.sqrt()`).
 - Indexado booleano y elegante para filtrar datos.
 - Guardar y cargar archivos con `np.save()` y `np.load()`.
 - Exportación de arreglos como `.csv` con `np.savetxt()`.
- **Pandas:**
 - Crear Series desde listas y diccionarios.
 - Crear DataFrames a partir de diccionarios.
 - Consultar propiedades como `dtypes`, `columns`, `index`.
 - Operaciones básicas: ordenamiento (`sort_values`, `sort_index`), transposición (`.T`), filtrado con `.isin()` y `.drop()`.
 - Exportar e importar archivos `.csv` con `to_csv()` y `read_csv()`.

Actividades sugeridas

- **Actividad inicial:** Mostrar la imagen del agujero negro y pedir a los estudiantes que investiguen qué papel jugó Python en su creación.
- **Taller con Numpy:**
 - Crear un arreglo de 20 elementos.
 - Modificar valores usando indexado booleano.
 - Calcular raíz cuadrada y exponencial con funciones universales.
- **Ejercicio práctico con Pandas:**
 - Crear un DataFrame con datos ficticios de estudiantes (nombre, carrera, promedio).
 - Ordenar por promedio, filtrar los de cierta carrera.
 - Guardar en archivo `.csv` y volver a cargarlo.
- **Desafío final:** Cargar un archivo `.csv` real (puede ser proporcionado por el profesor), limpiar los datos con Pandas y calcular estadísticas básicas (media, mínimo, máximo por columna).

Tema 16 Librerías para la visualización de los datos en Python

Enfoque para abordar la clase

Inicia la clase destacando **la importancia de la visualización de datos** como una herramienta clave para interpretar grandes volúmenes de información de manera clara. Introduce las librerías **Matplotlib, Seaborn y Plotly** como herramientas complementarias: Matplotlib como base sólida y versátil, Seaborn como una extensión estética y funcional, y Plotly como una opción moderna e interactiva. Explica que estas herramientas son ampliamente utilizadas en **ciencia de datos, inteligencia artificial y análisis exploratorio**. Utiliza una progresión didáctica: comienza con gráficos simples (líneas, puntos), luego estadísticas con Seaborn, y finaliza con la interactividad de Plotly para captar el interés visual de los estudiantes.

Ejemplos recomendados

- **Matplotlib:**
 - Trazar una línea simple (`plot()`) y ajustar el gráfico con `xlabel()`, `ylabel()`, `tick_params()`, etc.
 - Generar automáticamente 1000 puntos con `range()` y listas por comprensión.
 - Uso de `scatter()` para diagramas de dispersión.
 - Guardar un gráfico con `plt.savefig()`.
- **Seaborn:**
 - Cargar datasets con `sns.load_dataset()` y trazar gráficos con `relplot()` y `jointplot()`.
 - Mostrar relaciones multivariantes (e.g., propinas por tamaño de grupo, tiempo, fumador/no fumador).
 - Analizar correlaciones visuales con color y forma (`hue`, `style`, `size`).
- **Plotly:**
 - Crear una clase Dado y simular 100 lanzamientos.
 - Contar frecuencias y graficarlas con `Bar()` y `offline.plot()`.
 - Guardar la visualización como archivo HTML interactivo (`Dado.html`).

Actividades sugeridas

- **Inicio:** Muestra distintos tipos de gráficos (estáticos e interactivos) y pregunta a los estudiantes cuándo usarían cada uno.
- **Actividad guiada:**
 - Graficar una función cuadrática con Matplotlib y ajustar su diseño (título, ejes, grosor).
 - Usar Seaborn para representar visualmente un dataset conocido como tips o penguins.
 - Realizar un análisis conjunto con `jointplot()` para explorar correlaciones.
- **Desafío de cierre:**

- Simular 500 lanzamientos de un dado de 8 caras con una clase Dado, contar frecuencias y graficarlas con Plotly.
- Pedir a los estudiantes modificar el número de lados y lanzamientos, comparar gráficamente los resultados, y reflexionar sobre la ley de los grandes números.

Tema 17 Aplicaciones básicas de la inteligencia artificial

Enfoque para abordar la clase

Comienza presentando la inteligencia artificial (IA) como **un motor de transformación tecnológica y social**, haciendo énfasis en su papel dentro de la **cuarta revolución industrial**. Es importante resaltar que aunque la IA ya forma parte de nuestra vida cotidiana, **su comprensión profunda sigue siendo un reto técnico y ético**. Aborda la clase desde casos reales (como el sistema de predicción de hipoglucemia de IBM o los vehículos autónomos), promoviendo una reflexión crítica entre los estudiantes: ¿hasta qué punto la IA puede o debe sustituir funciones humanas? Organiza el contenido en dos bloques: **aplicaciones industriales y empresariales** (automatización, vehículos, robótica) y **tecnologías de lenguaje** (PLN, chatbots), permitiendo a los estudiantes identificar las áreas de impacto y discutir implicaciones sociales.

Ejemplos recomendados

- **IA en la industria:**
 - Sistema de predicción de hipoglucemia usando aprendizaje automático (bosque aleatorio).
 - Mantenimiento predictivo con sensores y redes neuronales en líneas de producción.
 - Aplicaciones en vehículos autónomos: comunicación V2V, integración IoT, y toma de decisiones.
- **Procesamiento de lenguaje natural (PLN):**
 - Caso Watson de IBM y su transición del entretenimiento a la medicina.
 - Análisis de emociones y sentimientos en redes sociales.
 - Detección de intenciones y entidades en un chatbot (por ejemplo, solicitud de horario bancario).
- **Agentes virtuales (chatbots):**
 - Diferencia entre intenciones, entidades y diálogo.
 - Variables de contexto para manejar flujos conversacionales complejos.
 - Analítica del comportamiento del usuario para optimización de respuestas.

Actividades sugeridas

- **Inicio:** Presenta una imagen o video corto de un vehículo autónomo o un chatbot en acción y plantea la pregunta: *¿qué tipo de tareas crees que realiza la IA para funcionar correctamente?*
- **Actividad de análisis grupal:**
 - Divide al grupo en tres equipos y asigna un dominio (industria, salud, atención al cliente). Pide que identifiquen un caso de uso de IA en ese contexto, sus beneficios, riesgos y cómo mejorarían la solución.
- **Actividad práctica:**
 - Simula una conversación de atención al cliente en donde el grupo deba identificar la intención y las entidades presentes. Luego construyan un diagrama de flujo simple de diálogo para un chatbot que resuelva esa consulta.
- **Reflexión de cierre:**
 - Discute con el grupo: *¿Qué profesiones crees que están más en riesgo por la IA? ¿Qué habilidades seguirán siendo exclusivamente humanas?*

Tema 18 Soluciones y tecnologías para hacer inteligencia artificial

Enfoque para abordar la clase

Aborda esta clase como una **exploración de las herramientas tecnológicas clave** que hacen posible la implementación de inteligencia artificial (IA) en el mundo real. Enfoca la sesión en presentar a los estudiantes una **visión panorámica del ecosistema de desarrollo de IA**, desde las bibliotecas de programación hasta las plataformas en la nube que facilitan su despliegue a gran escala. Utiliza una narrativa comparativa (pasado vs. presente) para explicar cómo la computación evolucionó desde los mainframes hasta la nube y cómo esta transición habilitó la democratización de la IA. Recalca el papel de la nube en la **escalabilidad, accesibilidad y colaboración**, así como los desafíos éticos y técnicos que aún persisten. Divide el tema en bloques: *marcos de trabajo, plataformas comerciales, cómputo en la nube y tendencias actuales*, promoviendo siempre una visión crítica sobre la selección de tecnologías y sus implicaciones.

Ejemplos recomendados

- **Marcos de trabajo de IA:**
 - Comparar TensorFlow, PyTorch y Keras en cuanto a complejidad, curva de aprendizaje y casos de uso típicos.
 - Ejemplo de un modelo de red neuronal simple construido con Keras.
- **Plataformas comerciales de IA:**
 - Presentar casos reales o demostraciones básicas de IBM Watson Studio, Google Cloud AI o Azure Machine Learning.

- Comentar las diferencias entre trabajar con una plataforma *on-premise* vs. una en la nube.
- **Modelos de servicio en la nube:**
 - Comparar SaaS, PaaS e IaaS con ejemplos cotidianos: Gmail (SaaS), Google App Engine (PaaS), Amazon EC2 (IaaS).
 - Mostrar la interacción entre estos modelos y el ciclo de vida de un proyecto de IA.
- **Tendencias actuales:**
 - Mostrar brevemente qué es AutoML con herramientas como H2O.ai o DataRobot.
 - Ejemplo de una aplicación con IA explicable (XAI) con visualización de pesos o decisiones del modelo.

Actividades sugeridas

- **Inicio:** Presenta una línea de tiempo breve que conecte el surgimiento de mainframes, la nube y las plataformas de IA. Plantea la pregunta: *¿Qué aspectos técnicos o sociales permitieron que la IA se expandiera en los últimos 10 años?*
- **Actividad de análisis comparativo:**
 - Divide al grupo en tres equipos: uno investigará plataformas de código abierto (ej. TensorFlow, Keras), otras plataformas comerciales (ej. Azure, IBM), y otros modelos de nube (SaaS, PaaS, IaaS).
 - Cada equipo debe presentar una recomendación para un proyecto ficticio de IA, justificando qué herramientas elegirían y por qué.
- **Actividad práctica (si el tiempo lo permite):**
 - Mostrar una demo sencilla en Google Colab con una red neuronal básica en Keras o utilizar AutoML de Google para clasificar imágenes o texto.
- **Reflexión de cierre:**
 - Plantear la pregunta: *¿Qué desafíos éticos y técnicos deben considerarse al usar plataformas de IA?*
 - Invitar a discutir la importancia del sesgo algorítmico, la explicabilidad y la soberanía de los datos.

Tema 19 Aplicaciones conversacionales básicas

Enfoque para abordar la clase

Presenta este tema como una **puerta de entrada al diseño de sistemas interactivos inteligentes**, enfatizando cómo los agentes conversacionales están cambiando la forma en que las personas interactúan con la tecnología. Comienza explicando la evolución histórica de los chatbots y su papel actual en servicios de atención al cliente, automatización y productividad. Resalta el valor del

procesamiento del lenguaje natural (PLN) como pilar técnico de estas soluciones. Es recomendable **dividir la sesión en dos partes**: una conceptual, enfocada en la arquitectura de un chatbot (componentes, flujo, contexto e intenciones), y una práctica, enfocada en la implementación básica con IBM Watson Assistant. Motiva a los estudiantes a visualizar estos sistemas como herramientas aplicables en múltiples industrias y proyectos personales.

Ejemplos recomendados

- **Casos reales:**
 - Chatbots usados en bancos, aerolíneas y tiendas en línea para reservar servicios, hacer pagos o resolver dudas comunes.
 - Alexa, Siri y Google Assistant como ejemplos de asistentes por voz con estructuras similares en cuanto a reconocimiento de intención y contexto.
- **Ejemplos didácticos:**
 - Frases como “¿Cuál es su capital?” tras “¿Conoces Italia?” para ilustrar la necesidad de seguimiento de contexto.
 - Mostrar cómo el chatbot puede fallar si no se configuran bien las entidades o los nodos.
- **Arquitecturas comparadas:**
 - Comparar la arquitectura básica de un chatbot de flujo basado en reglas vs. uno basado en aprendizaje automático, como el de Watson.

Actividades sugeridas

- **Inicio:** Haz una lluvia de ideas con la clase preguntando: *¿Con qué chatbots o asistentes han interactuado? ¿Qué les gustó o no les gustó de esa experiencia?*
- **Análisis guiado:**
 - Expón el diagrama de arquitectura del chatbot (figura 2 del material).
 - Pide a los alumnos que identifiquen y expliquen en equipo cada uno de los componentes (intenciones, contexto, diálogo, recuperación de información, etc.).
- **Práctica guiada:**
 - Si se dispone de conexión a internet y tiempo suficiente, realiza una práctica en vivo creando un chatbot básico con Watson Assistant (saludo y despedida).
 - Alternativamente, entrega capturas del entorno de IBM Cloud y plantea una actividad tipo “elige tu propia aventura” donde los alumnos diseñen el flujo conversacional de un asistente para una cafetería.
- **Cierre reflexivo:**
 - Pregunta a los estudiantes: *¿En qué proyecto personal, profesional o comunitario les gustaría implementar un chatbot?*
 - Introduce la idea de dominios verticales (atención médica, turismo, educación) y promueve la reflexión sobre la personalización de asistentes según el contexto.

Tema 20 Uso de aplicaciones conversacionales

Enfoque para abordar la clase

Plantea esta sesión como un **análisis aplicado del valor real** que aportan los chatbots en entornos comerciales. Enfócate en que el estudiantado entienda que más allá de una moda tecnológica, estas aplicaciones representan soluciones tangibles a problemáticas comunes: atención al cliente, automatización de pedidos, marketing conversacional, etc. Aborda la clase con un enfoque de **proyecto**, invitando a los estudiantes a reflexionar sobre cómo un chatbot puede resolver un caso específico. Resalta también los retos éticos y las malas prácticas asociadas al mal uso de datos o la generación de sesgos. Esta clase puede ser una excelente **transición del diseño conceptual al desarrollo funcional**.

Ejemplos recomendados

- **Caso de Domino's Pizza:** muestra cómo una marca puede integrar múltiples canales (Messenger, Alexa, Tinder) y lograr impacto comercial real con campañas conversacionales.
- **Chatbots por sector:** usa la gráfica del impacto en diferentes industrias para analizar con la clase qué sectores conocen o imaginan usando estos sistemas.
- **Pizzería "El horno mexicano":** este ejemplo práctico puede servir como base para ejercicios o desarrollo de proyectos en clase.

Actividades sugeridas

- **Discusión inicial:** pregunta "¿En qué momento del día interactuaron por última vez con un bot? ¿La experiencia fue positiva o negativa?" Esto puede abrir una reflexión sobre lo que esperan los usuarios de estas herramientas.
- **Análisis de caso:** divide al grupo en equipos y asígnales industrias distintas (banca, salud, educación, logística). Pídeles que identifiquen un problema real en ese sector que podría resolverse con un chatbot, justificando su respuesta con las 3 preguntas clave del texto.
- **Diseño guiado:** con base en el caso de "El horno mexicano", haz que los alumnos diseñen una variante (por ejemplo, una librería, cafetería o tienda de ropa) aplicando el mismo enfoque: intenciones, entidades y diálogo.
- **Reflexión ética:** plantea una lluvia de ideas sobre riesgos y malas prácticas en el uso de bots. Cierra con un debate sobre cómo diseñar bots responsables, sin sesgos, ni recopilación abusiva de datos.
- **Evaluación con métricas:** revisa las métricas de éxito (tasa de abandono, tasa de contención, latencia de respuesta) y propón una actividad tipo "analista": ¿cómo interpretarían esos resultados en una empresa real?

Tema 21 Campos de aplicación de la inteligencia artificial

Enfoque para abordar la clase

Plantea esta clase como una **exploración de la diversidad de aplicaciones reales de la inteligencia artificial**, más allá del ámbito técnico. Muestra cómo la IA ya está generando impacto en áreas tan distintas como el marketing, la salud, la educación o la visión computacional. Inicia con una pregunta guía: *¿en qué aspectos de tu vida diaria crees que ya interactúas con IA sin darte cuenta?* Aprovecha esa reflexión para motivar el interés del grupo. Este tema también puede servir para **fomentar el pensamiento crítico** respecto al impacto social y ético del uso masivo de estas tecnologías. Sugiere ver este tema como una oportunidad para que cada estudiante comience a identificar campos donde pueda aplicar la IA en su propio entorno.

Ejemplos recomendados

- **Caso Guo y Wu (2019)**: muestra cómo combinar visión por computadora y agentes conversacionales para construir soluciones de recomendación de productos más personalizadas y naturales.
- **BlueDot y Baidu frente al COVID-19**: ejemplifica el poder del aprendizaje automático para la predicción, diagnóstico y respuesta en contextos de salud pública.
- **Clasificación de imágenes** (Rosebrock, 2017): útil para explicar cómo los algoritmos interpretan visualmente una imagen en términos numéricos y probabilísticos.
- **Clasificadores personalizados de IBM**: excelente para mostrar cómo se puede entrenar un sistema específico según las necesidades del usuario final.

Actividades sugeridas

- **Lluvia de ideas**: organiza una dinámica en la que los estudiantes propongan sectores (educación, salud, industria, entretenimiento, etc.) y piensen en aplicaciones reales o potenciales de IA. Luego, contrasten con ejemplos reales presentados en clase.
- **Taller de clasificación de imágenes**: presenta un conjunto pequeño de imágenes y haz que los alumnos simulen cómo etiquetaría una IA (por ejemplo, fruta/no fruta), explicando las dificultades que enfrentaría un algoritmo frente a superposición o similitud visual.
- **Estudio de caso colaborativo**: divide a la clase en equipos y asigna a cada uno un caso del texto (Domino's, Baidu, Trunk, BlueDot). Cada equipo debe explicar el uso de IA, su impacto y una posible mejora que sugerirían al sistema.
- **Reflexión crítica**: pregunta al grupo: *¿En qué aspectos es positiva la automatización mediante IA? ¿Dónde puede representar un riesgo o desafío?* Esto promueve pensamiento ético y responsable sobre el diseño y uso de estas tecnologías.

- **Mini proyecto:** pide a los estudiantes que diseñen conceptualmente una aplicación de IA para resolver una problemática local o cotidiana, especificando qué datos usarían y qué tipo de algoritmo (clasificación, predicción, etc.).

Tema 22 Clasificadores básicos de imágenes

Enfoque para abordar la clase

En esta sesión, guía al grupo desde lo conceptual hacia lo práctico. Inicia con una breve historia de la visión por computadora para conectar con los orígenes de los clasificadores, destacando los avances desde el perceptrón hasta las redes neuronales profundas. Después, plantea el objetivo de la clase: **entender y aplicar los pasos clave para construir un clasificador de imágenes usando herramientas actuales como TensorFlow y Keras**. Recalca que no es necesario ser un experto para iniciarse en este campo, y que existen entornos amigables como Google Colab o Jupyter Notebook. Motiva a los estudiantes a ver este conocimiento como una forma de automatizar tareas visuales del mundo real.

Ejemplos recomendados

- **Reconocimiento de dígitos con MNIST:** útil para ilustrar los principios de clasificación, entrenamiento y evaluación con una red neuronal simple.
- **Caso Viola-Jones (detección de rostros en tiempo real):** ideal para explicar la transición de algoritmos clásicos a técnicas modernas basadas en aprendizaje profundo.
- **Amazon Rekognition o Azure Custom Vision:** como ejemplos de plataformas accesibles que los estudiantes pueden explorar para proyectos reales sin requerir código complejo.
- **Visualización con plt.imshow y predicciones con np.argmax:** buenos recursos para enseñar cómo interpretar los resultados de los modelos en un entorno práctico.

Actividades sugeridas

- **Actividad práctica guiada:** en Jupyter o Google Colab, lleva al grupo paso a paso para construir su propio clasificador de dígitos con el set MNIST (como en el ejemplo). Pide a los estudiantes modificar el número de épocas o capas para observar el impacto en la precisión.
- **Desafío rápido:** proporciona un conjunto diferente de imágenes (como ropa o animales simples) y haz que los estudiantes adapten el modelo a ese nuevo conjunto.
- **Comparación entre modelos:** divide la clase en equipos. Unos usarán un modelo con una sola capa oculta y otros con varias. Luego comparan exactitud, pérdida y tiempos de entrenamiento.
- **Análisis visual del sobreajuste:** muestra cómo el modelo puede ajustar demasiado los datos de entrenamiento y analiza con el grupo cómo la capa Dropout ayuda a evitarlo.
- **Debate breve:** plantea esta pregunta: *¿cuáles son los posibles riesgos o limitaciones de los clasificadores de imágenes en aplicaciones sensibles como salud o seguridad?*

Tema 23 Clasificadores básicos de imágenes

Enfoque para abordar la clase

Comienza la clase retomando brevemente el concepto de clasificación de imágenes, pero cambia el enfoque hacia sus aplicaciones reales. Recalca la importancia de la **visión artificial en contextos diversos**, como medioambiente, seguridad y moda. Esto ayudará al alumnado a visualizar el potencial práctico de los clasificadores más allá de los laboratorios. Luego, transita hacia el uso del conjunto **Fashion MNIST** para mostrar un ejemplo concreto y actual. La clase debe combinar reflexión sobre casos de uso con práctica técnica en Colab, resaltando cómo un mismo modelo base puede adaptarse a distintos dominios según el entrenamiento.

Ejemplos recomendados

- **Caso OmniEarth en California:** ilustra una aplicación medioambiental con impacto social. Úsalo para discutir cómo los clasificadores pueden apoyar políticas públicas.
- **VideoRecon de BlueChasm:** ejemplo útil para explicar cómo los clasificadores no se limitan a imágenes estáticas, sino que pueden aplicarse a video y audio.
- **Fashion MNIST y la moda inteligente:** excelente para conectar tecnología con intereses del alumnado. Comenta cómo estas técnicas ya están siendo utilizadas por plataformas de retail y recomendación.
- **Comparación visual de muestras predichas y reales con `predict_n_validate_individual_samples`:** permite entender errores del modelo y ajustar expectativas.

Actividades sugeridas

- **Proyecto guiado en Colab:** guía a los estudiantes paso a paso para construir un clasificador con Fashion MNIST (como el tutorial de Mohan). Asegúrate de explicar cada etapa: carga de datos, preprocesamiento, construcción del modelo, entrenamiento y evaluación.
- **Explora y modifica:** invita a los estudiantes a cambiar la arquitectura del modelo agregando una capa más o cambiando el número de neuronas. Luego comparen las métricas obtenidas.
- **Análisis de confusión:** propón que generen una matriz de confusión al final del modelo y discutan en grupo qué clases son más difíciles de predecir (por ejemplo, camisetas vs. blusas).
- **Discusión abierta:** plantea la pregunta "¿Qué desafíos éticos pueden surgir si estos clasificadores se aplican a datos personales como fotografías o grabaciones privadas?"
- **Mini debate:** divide al grupo y simula un escenario en el que una empresa quiere usar clasificación de imágenes para seguridad en el trabajo. Un equipo apoya la idea y otro la cuestiona.

Tema 24 Implementación de una aplicación de inteligencia artificial en la web

Enfoque para abordar la clase

Aborda esta clase como una **integración práctica** de los temas previamente vistos, especialmente en relación con el desarrollo y despliegue de modelos de inteligencia artificial. Inicia con una breve exposición de **Flatiron Health** para contextualizar el valor de llevar modelos a producción, especialmente en entornos críticos como el de la salud. Luego, traza una línea clara entre el desarrollo de modelos y su implementación real, destacando los desafíos comunes que enfrentan los desarrolladores en ese punto. Usa esta clase como una oportunidad para introducir el concepto de **DevOps** y su aplicación en entornos de IA, con énfasis en la automatización, el control de versiones y la colaboración. Procura mantener un enfoque demostrativo y guiado paso a paso, utilizando capturas de pantalla del entorno IBM Cloud o una demostración en vivo si es posible.

Ejemplos recomendados

- **Flatiron Health:** ideal para mostrar cómo la IA puede impactar positivamente en el sector salud. Úsalo como punto de partida para discutir la necesidad de llevar modelos funcionales al entorno clínico.
- **Diagrama del ciclo de vida DevOps:** ejemplifica cómo una aplicación puede evolucionar de un prototipo a un servicio estable. Ayuda a visualizar la continuidad de desarrollo y mantenimiento.
- **Caso de predicción de temperatura con vehículos USV:** excelente ejemplo por su simplicidad y aplicabilidad. Úsalo para demostrar el proceso completo: entrenamiento del modelo, despliegue y creación de interfaz web.

Actividades sugeridas

- **Demostración guiada:** realiza en clase el proceso de creación de un proyecto en IBM Watson Studio y despliegue de un modelo simple. Si el tiempo es limitado, al menos muestra cómo crear un espacio de trabajo y cargar un notebook.
- **Simulación de DevOps en equipo:** divide a los estudiantes en dos grupos: uno simula ser el equipo de datos que entrena el modelo y otro el equipo DevOps que se encarga del despliegue usando Toolchain. Deberán comunicarse entre sí para completar el proceso.
- **Mini reto: crear un endpoint funcional:** proporciona un modelo preentrenado y desafía a los estudiantes a desplegarlo en IBM Cloud con una prueba funcional desde Postman o un script en Python.
- **Discusión guiada:** plantea la pregunta: *¿Qué aspectos no técnicos deben considerarse al poner un modelo de IA en producción?* (ética, privacidad, rendimiento, mantenibilidad).

- **Explora Toolchain:** pide a los estudiantes que exploren la interfaz de Toolchain y propongan cómo podrían usarla para automatizar un flujo de integración continua en sus propios proyectos.



Fase I. Planteamiento

Descripción

Se realizará el planteamiento de una situación que será analizada y resuelta mediante la aplicación de herramientas de inteligencia artificial. En esta fase se creará el conjunto de datos base con los que trabajará en las futuras fases.

Objetivo

Detectar una problemática que pueda ser resuelta con la intervención de la inteligencia artificial y generar los datos necesarios para resolverla.

Requerimientos

- Saber identificar y clasificar datos según su tipo.
- Saber resolver problemas matemáticos utilizando el teorema de Bayes.
- Entender las fases que intervienen el proceso de aprendizaje automático.
- Comprender las implicaciones éticas del manejo y uso de los datos.

Instrucciones

1. Revisa:
 - Los tipos de paradigmas de la inteligencia artificial.
 - Los diferentes tipos de datos y su clasificación.
 - La descripción de la metodología CRISP-DM.
 - Las herramientas más adecuadas para las diferentes fases de la metodología CRISP-DM
2. Selecciona y justifica una problemática de tu interés, donde puedas identificar claramente todos los datos que la describen.

Ejemplo de proyectos:

- Evaluación del estado de salud de un grupo de personas.
 - Comportamiento del clima durante un periodo de tiempo.
 - Seguimiento del estado de una criptomoneda.
 - Estado de los vuelos internacionales y restricciones aplicadas por países.
 - Precio de los automóviles por temporadas.
3. Recopila la mayor cantidad de datos que puedas sobre la problemática seleccionada y realiza la clasificación de estos según su tipo. La cantidad de datos recopilados debe ser de un valor considerablemente alto.

4. Aplica la primera y segunda fase de la metodología ASUM-DM de IBM y define el tipo de información relevante que necesitas para resolver la problemática o situación.
5. Realiza un análisis a partir de los conocimientos adquiridos durante el módulo y determina cuál es la rama o aplicación de la inteligencia artificial más adecuada para la situación seleccionada.
6. Presenta tus conclusiones en un documento electrónico y compártela con tus compañeros mediante un foro. (Puede ser un material audiovisual en caso de ser requerido).

Criterios de evaluación

Criterio	Puntaje
1. Selecciona y demuestra que la problemática de interés cumple con los requisitos solicitados.	20
2. Recopila y clasifica los datos adquiridos.	23
3. Aplica las fases de la metodología y define el tipo de información que necesita para resolver el problema.	22
4. Selecciona la herramienta de inteligencia artificial adecuada.	20
5. Elabora el documento (incluyendo las conclusiones) y realiza la presentación oral de sus conclusiones.	15

Entregable

Documento digital con evidencias del proceso de recopilación de los datos y los resultados de la aplicación de las dos primeras fases de la metodología indicada. Incluye sus conclusiones personales sobre el análisis realizado.



Fase II. Ejecución

Descripción

En esta fase se realizará el procesamiento de los datos recopilados en la fase anterior, haciendo uso de las herramientas y conocimientos de programación en Python obtenidos durante el desarrollo del Módulo 2.

Objetivo

Analizar un conjunto de datos, utilizando el lenguaje de programación Python para obtener un reporte relevante de estos.

Requerimientos

- Comprender los conceptos generales del lenguaje Python.
- Preparar el entorno de trabajo con Jupyter Notebook.
- Comprender el uso de variables, estructuras de datos y funciones en Python.
- Saber operar con archivos externos y las librerías de manejo de datos y visualización de Python.
- Entender la programación orientada a objetos en Python.
- Tener el conjunto de datos recopilados de la Fase 1.

Instrucciones

1. Revisa:
 - Los conceptos generales de los lenguajes de programación y sus enfoques hacia a la inteligencia artificial.
 - El funcionamiento de los diccionarios y conjuntos en Python.
 - La forma de declarar funciones y combinarlas con las estructuras de control en Python.
 - La instalación y uso de las librerías NumPy, Pandas y Matplotlib en Python.
2. Agrupa los datos recopilados en un fichero tipo texto que pueda ser importado a Python, de preferencia "csv" (Comma Separated Values) o "xlsx" (Microsoft Excel Spreadsheet as Data Charts).
3. Prepara tu entorno de trabajo en Jupyter Notebook dentro de la plataforma IBM Cloud, puedes apoyarte en la guía compartida en el anexo 1.
4. Importa el fichero de datos al entorno de trabajo y realiza las siguientes operaciones:
 - Importa las librerías NumPy, Pandas y Matplotlib a tu notebook.
 - Genera un Dataframe con los datos incluidos en el fichero de datos obtenido en el punto 2.
 - Organiza la información dentro de Dataframe de la forma que mejor se adecue a la problemática que se desea analizar, modifica el tipo de variable en caso de ser necesario.
 - Aplica una función matemática que te permita sintetizar parte de los datos dentro del Dataframe.

- Selecciona una muestra de los datos y representa gráficamente la información que aportan.
5. Elabora un resumen de la información relevante obtenida, presenta tus conclusiones en un documento electrónico y compártela con tus compañeros en el foro indicado por el profesor (puede ser un material audiovisual en caso de ser requerido).

Criterios de evaluación

Criterio	Puntaje
1. Elabora el documento con los datos recopilados.	20
2. Prepara el entorno de trabajo en IBM Cloud.	20
3. Desarrolla las operaciones indicadas.	20
4. Obtiene información relevante de los datos analizados.	20
5. Elabora el documento y comparte sus conclusiones en el foro indicado por el profesor.	20

Entregable(s)

- Documento digital con evidencias del proceso de análisis de los datos y las conclusiones elaboradas a partir de la información relevante obtenida.
- Enlace al Notebook elaborado con la plataforma IBM Cloud.



Fase III. Resultados

Descripción

En esta fase se construirá una aplicación de inteligencia artificial, a partir de los datos recopilados y procesados en las fases anteriores, utilizando diferentes herramientas de la plataforma IBM Cloud y se realizará el despliegue de esta para ser consumida por servicios externos.

Objetivo

Construir y desplegar una aplicación de inteligencia artificial capaz de recibir datos y devolver información relevante de estos.

Requerimientos

- Comprender los fundamentos generales de programación en Python.
- Acceso a la plataforma IBM Cloud.
- Distinguir entre las diferentes aplicaciones que posee Watson Studio y sus principales usos.
- Comprender los conceptos de modelos, activos y despliegues dentro de un proyecto de Watson Studio.
- Interpretar cómo funcionan las API y la manera interactuar con ellas.
- Tener el conjunto de datos recopilados y procesados durante las fases anteriores.

Instrucciones

1. Revisa:
 - Las características principales de los Objects Storage, Watson Machine Learning y Watson Studio dentro de la plataforma IBM Cloud.
 - Importar y manipular datos en formato "csv" con Python dentro de los cuadernos de Jupyter.
 - Cómo funciona Watson AutoAI.
 - Los tipos de modelos de aprendizaje automático y sus respectivos modos de uso.

- Los conceptos de punto final de API, token de autorización de IBM Cloud y consumo general de APIs con Python.
2. Accede a la plataforma IBM Cloud con tu cuenta educativa y genera un nuevo proyecto de Watson Studio.
 3. Navega hasta la opción de activos y agrega un nuevo experimento de AutoAI a tu proyecto. Incorpora una nueva fuente de datos al proyecto e importa el archivo "csv" generado y procesado durante las fases anteriores. En la sección de configuración indícale que información deseas que devuelva una vez generado el modelo.
 4. En función del tipo de problemática que estás analizando, selecciona un modelo de inteligencia artificial que más se adecúe a tu caso. Recuerda que AutoAI puede utilizar los modelos de clasificación binaria, clasificación multiclase y regresión.
 5. Ejecuta el experimento y analiza las métricas de evaluación. Selecciona el modelo que muestre un mejor desempeño (considera de manera prioritaria los valores de exactitud y precisión) y guárdalo en tu proyecto principal.
 6. Crea un nuevo despliegue y comprueba la calidad de los resultados realizándole varias pruebas al modelo desplegado. Consulta los fragmentos de código que comparte la implementación para permitir el acceso desde fuentes externas a IBM Cloud.
 7. Crea un cuaderno de Jupyter en cualquier otra herramienta con la que sepas trabajar (Anaconda, Google Colab, entre otras), desarrolla un programa que consuma el API creada y que muestre los resultados de las consultas de una forma amigable.

Elabora un resumen de la información relevante obtenida durante el desarrollo de las actividades de la fase y presenta tus conclusiones en un documento electrónico.

Criterios de evaluación

Criterio	Puntaje
1. Genera el proyecto en Watson Studio y agrega el experimento de AutoAI.	10
2. Incorpora el conjunto de datos al experimento y realiza las configuraciones indicadas.	20

3. Ejecuta, analiza y selecciona el modelo que mejor desempeño mostró de acuerdo con las métricas entregadas.	20
4. Crea el despliegue solicitado, realiza las pruebas adecuadas y elabora el programa de comprobación utilizando Python y Jupyter Notebook.	30
5. Elabora el documento y realiza la presentación escrita de sus conclusiones.	20

Entregables

- Documento digital con evidencias del proceso de creación del proyecto, trabajo con la herramienta de Watson AutoAI, comprobación del modelo creado y las conclusiones elaboradas a partir de la información relevante obtenida.
- Enlace o copia del Notebook elaborado para comprobar el despliegue del modelo.

Prácticas adicionales

Las siguientes actividades no forman parte de la evaluación formal, pero se recomienda que las realicen los alumnos para asegurar el logro de la competencia.

Práctica adicional 1. El aprendizaje automático, una visión al futuro de la inteligencia artificial

Esta actividad abarca los temas 5 y 6.

Objetivo de la actividad

Identificar los tipos de inteligencia artificial que existen y sus diferentes formas de implementación.

1. Elabora un cuadro comparativo (mínimo cinco elementos) entre la inteligencia artificial simbólica y la neuronal.
2. Investigue tres aplicaciones actuales de inteligencia artificial simbólica en la industria y elabore una infografía donde exponga los resultados encontrados.
3. Investigue tres aplicaciones actuales del aprendizaje automático en la educación y un video educativo donde exponga los resultados encontrados.
4. Identifique dos situaciones problemáticas donde se pueda aplicar el teorema de Bayes para comprender mejor el fenómeno y presente sus conclusiones a partir de cálculos matemáticos concretos.
5. Elabore una tabla comparativa con los diferentes algoritmos que conforman el marco de trabajo del aprendizaje automático y sus principales aplicaciones.

Criterios de evaluación

- Elabora el cuadro comparativo.
- Elabora la infografía solicitada.
- Elabora el video sobre la temática sugerida.
- Presenta los resultados matemáticos de la aplicación del teorema de Bayes.
- Elabora una tabla comparativa con los diferentes algoritmos.

Entregable (s)

Documento electrónico con el desarrollo de la actividad.

Práctica adicional 2. La computación cognitiva y los retos futuros de la inteligencia artificial

Esta actividad abarca los temas 7 y 8.

Objetivo de la actividad

Identificar las principales aplicaciones de la computación cognitiva y comprender la responsabilidad ética del uso de la inteligencia artificial en la sociedad.

1. Elabore una tabla comparativa con los conceptos cognición, inteligencia artificial e inteligencia artificial cognitiva. Incluya tres semejanzas, tres diferencias y dos casos de uso.
2. Investigue tres casos donde se utilizó la inteligencia artificial con fines poco éticos y analice las consecuencias de estas prácticas en la sociedad.
3. Elabore una infografía donde exprese su proyección personal del impacto de la utilización de la inteligencia artificial en los próximos 50 años.

Criterios de evaluación

- Elabora la tabla comparativa.
- Realiza y demuestra los resultados de la investigación.
- Elabora la infografía requerida.

Entregable (s)

Documento electrónico con el desarrollo de la actividad.

Práctica adicional 3. Uso de librerías y herramientas para el manejo de datos y su visualización en Python

Esta actividad abarca los temas 15 y 16.

Objetivo de la actividad

Programar ejercicios que implementen las diferentes librerías y herramientas vistas para el manejo de datos.

Instrucciones

- Importa la librería Pandas a tu proyecto.

- Genera un diccionario con al menos 10 elementos, donde las estructuras llave-valor correspondan a diferentes ciudades de México con su respectiva densidad de población (ejemplo: Guadalajara: 1,521,741).
- Genera una *Series* a partir del diccionario anterior.
- Incluye tres nuevos índices dentro de la *Series*, correspondientes a otras ciudades de México, y verifica que Pandas autocompletó los campos con los valores "NaN" para mantener el ajuste del arreglo.
- Finalmente, elige alguna de las herramientas de visualización vistas en clase y realiza la gráfica de los datos.

Criterios de evaluación

Programas en Python funcionando adecuadamente.

Entregable (s)

Documento electrónico con el desarrollo de la actividad que incluya el archivo .ipynb con el código de ambos programas en Python.