

Unidad 1: Introducción a la programación básica y a la IA

Descripción del Curso

Este curso está diseñado para estudiantes de todas las edades y contextos, sin restricción de edad, que buscan desarrollar de forma integral habilidades cognitivas, sociales y éticas aplicables a situaciones reales. A lo largo de la asignatura, se fomentará un aprendizaje activo y participativo, donde el alumno pasa de la adquisición pasiva de conceptos a la construcción de conocimiento mediante experiencias prácticas, proyectos interdisciplinarios y reflexión continua. La metodología combina actividades colaborativas, resolución de problemas, debates, estudios de caso y uso de herramientas digitales para promover la autonomía, la creatividad y la responsabilidad ciudadana. La materia se organiza en unidades que enfatizan el pensamiento crítico, la comunicación efectiva y la capacidad de transferir lo aprendido a contextos variados de la vida diaria y futura trayectoria educativa o profesional. Se prioriza un enfoque inclusivo, la valoración de la diversidad y el desarrollo de hábitos de estudio sostenibles. Al finalizar el curso, el estudiante puede explicar conceptos clave, justificar decisiones con argumentos sólidos, colaborar en equipos, gestionar su aprendizaje y evaluar críticamente el impacto de sus acciones en su entorno.

Competencias

- Pensamiento crítico y creativo para analizar situaciones y proponer soluciones. - Comunicación oral y escrita clara, persuasiva y adecuada al público objetivo. - Trabajo colaborativo y habilidades de liderazgo y gestión de proyectos en equipo. - Alfabetización digital y uso responsable de la información y las fuentes. - Resolución de problemas y toma de decisiones fundamentadas en evidencia. - Ética, empatía y respeto a la diversidad, con responsabilidad cívica. - Aplicación de conocimientos a contextos reales y capacidad de transferir aprendizajes a nuevas situaciones.

Requerimientos

- No hay restricción de edad para el acceso al curso. - Interés activo por aprender y participar en actividades prácticas y debates. - Acceso a una plataforma educativa en línea y/o materiales impresos según la modalidad (presencial o virtual). - Disposición para trabajar en equipo, cumplir plazos y participar en evaluaciones formativas y sumativas. - Materiales básicos: cuaderno o bloc de notas y bolígrafo; dispositivo con conexión a Internet para tareas en línea (según modalidad) y lectura de materiales complementarios.

Unidades del Curso

Unidad 1: Unidad 1: Introducción a la programación básica y a la IA

Objetivos de Aprendizaje

- Definir variables y tipos de datos básicos con ejemplos simples (números, texto, booleanos).
- Describir qué es un agente y qué es su entorno en un contexto de IA.

- Explicar de forma breve cómo se puede modelar una tarea de IA con código sencillo.

Contenidos Temáticos

1. Tema 1: Variables y tipos de datos

Conceptos de almacenamiento de información y tipos básicos que usan los programas.

2. Tema 2: Operadores y expresiones simples

Cómo combinar valores con operadores para crear expresiones que el programa evalúa.

3. Tema 3: Agente y entorno en IA

Idea de agente que percibe un entorno y toma decisiones simples, como marco de modelado de tareas de IA.

Actividades

• Actividad 1: Exploración de variables en situaciones cotidianas

Los estudiantes identifican variables en ejemplos del día a día (edad, temperatura, estado verdadero/falso) y las representan en un pequeño esquema de datos. Puntos clave: identificar datos, elegir tipos adecuados y ver cómo cambian con el tiempo.

• Actividad 2: Construcción de expresiones simples

Se crean expresiones usando operadores para evaluar condiciones simples (p. ej., temperatura > 20 y luz encendida). Puntos clave: interpretación de resultados y relación con decisiones del programa.

• Actividad 3: Modelado de un agente y su entorno

Diagrama breve y pseudocódigo de un agente sencillo que percibe una señal y responde con una acción. Puntos clave: relación entre percepción, decisión y acción.

Evaluación

- Cuestionario corto de conceptos sobre variables, tipos de datos y operadores (alineado al objetivo general).
- Actividad de modelado: identificar conceptos de agente y entorno en un contexto sencillo y presentar un breve diagrama/pseudocódigo.
- Participación y precisión en las actividades prácticas sobre variables y operadores.

Unidad 2: Unidad 2: Lógica booleana y toma de decisiones en programas

Objetivos de Aprendizaje

- Definir operadores booleanos (AND, OR, NOT) y su uso en condiciones.
- Explicar cómo se evalúan condiciones en estructuras condicionales básicas.
- Aplicar la lógica booleana para decidir entre dos o más acciones en un ejemplo práctico.

Contenidos Temáticos

1. Tema 1: Lógica booleana y operadores

Conjunciones, disyunciones y negaciones; cómo se combinan para formar condiciones.

2. Tema 2: Condiciones en programas

Estructuras if/else y su relación con la lógica booleana.

3. Tema 3: Decisiones basadas en percepciones

Ejemplos de escenarios simples donde la decisión depende de percepciones del entorno.

Actividades

• Actividad 1: Juego de booleans

Resolver problemas cortos que requieren combinar condiciones con AND, OR y NOT para obtener una acción. Puntos clave: interpretación de condiciones y su impacto en la acción.

• Actividad 2: Construcción de decisiones con if

Crear decisiones simples en pseudocódigo o en un lenguaje de aula que evalúen percepciones (ej. temperatura, energía de sensores) y elijan una acción correspondiente.

• Actividad 3: Taller de análisis de casos

Analizar diferentes escenarios y justificar por qué la lógica booleana produce una determinada decisión.

Evaluación

- Ejercicios escritos de evaluación de condiciones booleanas y su traducción a código simple.
- Actividad de diseño de una pequeña decisión de agente basada en percepciones booleanas.
- Participación y claridad al explicar la relación entre lógica booleana y decisiones de programas.

Unidad 3: Unidad 3: Pseudocódigo para soluciones simples

Objetivos de Aprendizaje

- Definir qué es pseudocódigo y sus ventajas para planificar soluciones.
- Escribir secuencias de pasos lógicos para resolver un problema sencillo.
- Modelar decisiones y repeticiones mediante pseudocódigo (condicionales y bucles).

Contenidos Temáticos

1. Tema 1: Qué es pseudocódigo y cómo se escribe

Convenciones simples para expresar ideas sin sintaxis de un lenguaje concreto.

2. Tema 2: Secuencias y soluciones paso a paso

Orden lógico de acciones para alcanzar un objetivo.

3. Tema 3: Condicionales y bucles en pseudocódigo

Modelar decisiones y repeticiones de forma clara y legible.

Actividades

- **Actividad 1: Esquema de solución en pseudocódigo**

Planificar la solución de una tarea simple (p. ej., decidir qué acción tomar en función de dos percepciones) en pseudocódigo. Puntos clave: claridad, orden y correspondencia con el problema.

- **Actividad 2: Transcripción de un diagrama de flujo**

Convertir un diagrama de flujo sencillo a pseudocódigo y justificar las decisiones tomadas.

- **Actividad 3: Revisión entre pares**

Intercambiar pseudocódigos con un compañero y proponer mejoras para mayor claridad y precisión.

Evaluación

- Ejercicios de escritura de pseudocódigo para problemas simples; evaluación de la lógica y estructura.
- Actividad de conversión de diagramas de flujo a pseudocódigo y justificación de decisiones.
- Retroalimentación entre pares sobre claridad y exactitud del pseudocódigo.

Unidad 4: Unidad 4: Programación básica en Python: decisiones de un agente

Objetivos de Aprendizaje

- Utilizar variables para almacenar percepciones y acciones del agente.
- Implementar decisiones simples con estructuras if/else en Python.
- Leer entradas básicas y mostrar salidas relevantes que describan la acción del agente.

Contenidos Temáticos

1. Tema 1: Estructura básica de Python

Variables, impresión y lectura de datos; sintaxis básica para empezar a programar.

2. Tema 2: Estructuras de decisión if/else

Cómo decidir entre acciones en función de percepciones.

3. Tema 3: Modelando un agente con percepciones y acciones

Construcción de un pequeño agente que toma decisiones basadas en entradas simples.

Actividades

- **Actividad 1: Programa decisonal básico en Python**

Escribir un programa que, dadas percepciones simples (p. ej., temperatura y luz), elija una acción (p. ej., encender/acondicionar). Puntos clave: uso de variables y if/else.

- **Actividad 2: Ampliar percepciones y acciones**

Agregar una segunda percepción y una segunda rama de acción para enriquecer el comportamiento del agente.

- **Actividad 3: Verificación y lectura de salidas**

Ejecutar el programa con diferentes entradas y explicar por qué el agente elige cada acción.

Evaluación

- Evaluación de un programa Python funcional que implemente una decisión de agente basada en condiciones simples.
- Rúbrica de claridad del código, correcta utilización de variables y manejo básico de entradas/salidas.
- Explicación oral o escrita de la lógica detrás de las decisiones del agente.

Unidad 5: Estructuras de control de flujo: if/else y bucles

Objetivos de Aprendizaje

- Crear condicionales anidados y expresiones lógicas para decisiones complejas.
- Implementar bucles para repetir acciones hasta alcanzar un objetivo.
- Aplicar estructuras de control de flujo en soluciones simples de agentes.

Contenidos Temáticos

1. Tema 1: Condicionales avanzados

Condiciones anidadas, uso de operadores lógicos y control de flujo dentro de bloques if.

2. Tema 2: Bucles básicos

While y for para repetición de acciones y verificación de condiciones de terminación.

3. Tema 3: Aplicaciones en agentes simples

Diseño de soluciones donde el agente repite acciones o toma decisiones en bucles controlados.

Actividades

- **Actividad 1: Condicionales anidados**

Diseñar un conjunto de reglas para un agente que debe decidir entre varias acciones según dos percepciones y su estado. Puntos: claridad de condiciones y respuestas adecuadas.

- **Actividad 2: Bucle de simulación**

Implementar un bucle que repita acciones hasta alcanzar un objetivo simple (p. ej., acumular puntos o cumplir una condición de entorno).

- **Actividad 3: Depuración de flujo**

Detectar y corregir errores comunes de flujo en un programa de agente con if/else y bucles.

Evaluación

- Actividad de implementación de condicionales y bucles en un programa corto que simule un agente.
- Evaluación de la capacidad de explicar la secuencia de decisiones y la terminación de bucles.
- Revisión entre pares de la legibilidad y la corrección lógica del flujo del programa.

Unidad 6: Estructuras de datos simples: listas para percepciones y acciones

Objetivos de Aprendizaje

- Crear y usar listas para almacenar percepciones y acciones del agente.
- Acceder, modificar, agregar y eliminar elementos de listas.
- Iterar sobre listas para producir secuencias de acciones del agente.

Contenidos Temáticos

1. Tema 1: Introducción a listas

Estructuras de datos básicas y operaciones comunes (append, acceso por índice).

2. Tema 2: Listas para percepciones y acciones

Almacenamiento de percepciones recibidas y acciones posibles o realizadas.

3. Tema 3: Manipulación de listas en contextos de agente

Ejemplos de procesamiento de percepciones en listas para decidir acciones.

Actividades

- **Actividad 1: Construir listas de percepciones**

Crear una lista de percepciones simuladas y practicar operaciones básicas (append, indexado, slicing).

- **Actividad 2: Generar acciones a partir de listas**

Utilizar una lista de percepciones para decidir una acción, iterando sobre ella y registrando acciones en otra lista.

- **Actividad 3: Depuración de manejo de listas**

Resolver ejercicios que involucren acceso fuera de rango o modificaciones no deseadas de listas.

Evaluación

- Proyecto corto en el que el agente lea percepciones de una lista y registre las acciones resultantes en otra lista.
- Demostración de manejo correcto de listas: inserción, acceso y recorrido.
- Explicación de la relación entre percepciones almacenadas y las decisiones del agente.

Unidad 7: Unidad 7: Depuración de errores y buenas prácticas

Objetivos de Aprendizaje

- Identificar errores de sintaxis comunes y sus mensajes de error.
- Detectar errores de lógica y justificar por qué el programa no se comporta como se espera.
- Aplicar estrategias simples de depuración (imprimir estados, pruebas por pasos, pruebas con entradas variadas).

Contenidos Temáticos

1. Tema 1: Errores de sintaxis

Errores por faltas de puntuación, nombres y estructura del código.

2. Tema 2: Errores de lógica

Errores en condiciones, bucles y asignaciones que producen resultados incorrectos.

3. Tema 3: Pruebas y depuración

Estrategias simples para probar y corregir programas de forma sistemática.

Actividades

• Actividad 1: Identificar y corregir errores

Se proporcionan fragmentos de código con errores intencionados; los estudiantes deben identificarlos y corregirlos.

• Actividad 2: Pruebas simples

Diseñar casos de prueba para verificar comportamientos esperados y registrar resultados.

• Actividad 3: Revisión de pares

Intercambiar código entre compañeros y proponer mejoras de legibilidad y robustez.

Evaluación

- Resolución de ejercicios de depuración, demostrando la corrección de errores y la capacidad de justificar las soluciones.
- Informe corto sobre técnicas de depuración aplicadas y resultados observados.
- Participación y trabajo en equipo en las actividades de revisión entre pares.

Unidad 8: Unidad 8: Conceptualización de un agente de IA y límites de la IA básica

Objetivos de Aprendizaje

- Describir, en palabras simples, cómo un agente utiliza percepciones para decidir una acción.
- Identificar limitaciones de un agente básico (falta de aprendizaje, adaptabilidad, generalización).

- Proponer mejoras simples para ampliar el comportamiento de un agente básico (p. ej., incorporar más percepciones o reglas).

Contenidos Temáticos

1. Tema 1: Simulación de un agente simple

Resumen de cómo se percibe el entorno y se toma una acción básica.

2. Tema 2: Evaluación de percepciones para la acción

Relación entre percepciones recibidas y la acción elegida.

3. Tema 3: Límites de IA básica y comparación con IA avanzada

Discusión sobre qué puede y qué no puede hacer un agente básico frente a enfoques más complejos (aprendizaje, adaptabilidad, generalización).

Actividades

• Actividad 1: Mapa conceptual de un agente simple

Crear un mapa conceptual que conecte percepciones, reglas de decisión y acciones, destacando limitaciones.

• Actividad 2: Debate sobre límites

Debate estructurado sobre cuándo un agente básico es suficiente y cuándo se requieren enfoques más avanzados de IA.

• Actividad 3: Proyecto corto de agente

Diseñar un mini-proyecto en el que el agente perciba un entorno y tome acciones simples, describiendo sus limitaciones y posibles mejoras.

Evaluación

- Presentación breve (oral o escrita) explicando el funcionamiento conceptual de un agente simple.
- Análisis de límites y posibles mejoras en un informe corto.
- Participación en el debate y calidad de las propuestas de mejora.