

Unidad 1: Fundamentos de Programación y IA: conceptos y entorno

Descripción del Curso

Esta unidad, integrada en un curso sobre fundamentos de Inteligencia Artificial para estudiantes, aborda de manera conceptual cómo un programa básico puede simular un agente de IA evaluando percepciones del entorno y eligiendo acciones. Se exploran las limitaciones de este enfoque frente a métodos más avanzados, como el aprendizaje y la planificación, para que el estudiante entienda cuándo un agente simple es suficiente y cuándo se requieren estrategias más complejas. Al finalizar la unidad, el estudiante podrá explicar a nivel conceptual cómo un programa básico simula un agente de IA evaluando percepciones y seleccionando acciones, y señalar las limitaciones respecto a enfoques más complejos. Los objetivos específicos incluyen describir el ciclo de percepción-decisión-acción de un agente básico; comparar un agente básico con enfoques de IA más avanzados (aprendizaje, planificación, incertidumbre); e ilustrar con un ejemplo sencillo cómo se evalúan percepciones y se elige una acción. Esta unidad promueve el desarrollo del pensamiento crítico, la capacidad de explicar conceptos tecnológicos con claridad y la habilidad para analizar situaciones cotidianas en las que un agente simple toma decisiones, conectando estos conceptos con implicaciones prácticas y éticas de la IA en tareas simples y reales.

Competencias

- Comprender de forma conceptual el ciclo de percepción-decisión-acción de un agente básico y su evaluación de percepciones.
- Analizar y comparar un agente básico con enfoques de IA más avanzados (aprendizaje, planificación, manejo de incertidumbre).
- Aplicar conceptos a ejemplos simples para describir cómo se evalúan percepciones y se elige una acción.
- Desarrollar pensamiento crítico sobre las limitaciones de los agentes simples y sus aplicabilidades en contextos reales.
- Comunicar ideas sobre IA con claridad, usando terminología adecuada y ejemplos comprensibles.
- Colaborar en actividades prácticas y presentar soluciones simples a problemas de IA en escenarios cotidianos.

Requerimientos

- Interés en ciencias de la computación y fundamentos de IA.
- Conocimientos básicos de matemática y lógica básica para comprender secuencias de decisión.
- Capacidad para leer e interpretar textos técnicos y ejemplos sencillos.
- Acceso a una computadora o dispositivo con conexión a internet para ejercicios y simulaciones simples.
- Disponibilidad para participar en discusiones, debates y actividades prácticas en clase o en modalidad virtual.

- Materiales: cuaderno, bolígrafo o lápiz, y acceso a herramientas o simuladores propuestos por el docente (opcionales según la metodología).

Unidades del Curso

Unidad 1: Unidad 1: Fundamentos de Programación y IA: conceptos y entorno

Objetivos de Aprendizaje

- Identificar variables y tipos de datos en ejemplos simples de programación.
- Describir qué es un agente y qué es un entorno en un problema de IA.
- Explicar con ejemplos cómo se modela una tarea de IA en código sencillo.

Contenidos Temáticos

1. Conceptos de programación básica: Descripción corta

1. Variables y tipos de datos
2. Operadores y expresiones

2. IA: agente y entorno: Descripción corta

1. Definición de agente
2. Definición de entorno

Actividades

- **Actividad 1: Explorando conceptos básicos** - Identificar variables, tipos de datos y operadores en ejemplos simples; explicar con sus propias palabras qué representa cada elemento y cómo influye en la ejecución de un programa. Puntos clave: representar datos, elegir tipos adecuados, entender operaciones básicas y su efecto en la salida. Aprendizajes: reconocimiento de conceptos y capacidad de describir su función en un programa.
- **Actividad 2: ¿Qué es un agente y qué es un entorno?** - Trabajo en grupo para definir un problema sencillo (por ejemplo, un cajero automático o un juego simple) en términos de agente y entorno, y crear una breve representación en palabras o diagrama.
- **Actividad 3: Modelando una tarea de IA con código** - Construir un ejemplo corto de IA simbólica (sin aprendizaje) que demuestre cómo el agente percibe el entorno y decide una acción basada en condiciones simples. Aprendizajes: entender el flujo de percepción ? decisión ? acción en una tarea de IA.

Evaluación

- Rúbrica para identificar conceptos: clasificación correcta de variables, tipos de datos y operadores (20%).
- Explicación conceptual de agente y entorno (40%).
- Actividad práctica de modelado de una tarea de IA con código básico (40%).

Unidad 2: Unidad 2: Lógica booleana y decisiones en programación

Objetivos de Aprendizaje

- Explicar qué es la lógica booleana y cómo se aplica en condicionales simples.
- Distinguir entre condiciones verdaderas y falsas y su impacto en el flujo del programa.
- Ilustrar con ejemplos cómo una decisión depende de percepciones del entorno.

Contenidos Temáticos

1. Lógica booleana y operadores lógicos (and, or, not): Descripción corta

1. Conceptos booleanos
2. Operadores lógicos

2. Condicionales y control de flujo: Descripción corta

1. Estructuras if/else
2. Evaluación de condiciones

Actividades

- **Actividad 1: Analizando condiciones** - Actividad de discusión y ejercicios cortos para identificar condiciones verdaderas vs falsas y predecir el flujo del programa.
- **Actividad 2: Simulación de decisiones** - Construcción de ejemplos simples con if/else que representen decisiones ante percepciones de un entorno ficticio.
- **Actividad 3: Pautas de evaluación de condiciones** - Ejercicios de razonamiento para justificar por qué una condición se evalúa como verdadera o falsa.

Evaluación

- Capacidad para justificar decisiones basadas en condiciones booleanas (30%).
- Identificación de errores comunes en condicionales (30%).
- Aplicación de condicionales en un mini-ejercicio de código (40%).

Unidad 3: Unidad 3: Pseudocódigo para soluciones simples

Objetivos de Aprendizaje

- Escribir secuencias de pasos para resolver problemas simples en pseudocódigo.
- Representar decisiones mediante estructuras condicionales en pseudocódigo.
- Incorporar bucles para repetir acciones cuando sea necesario.

Contenidos Temáticos

1. Introducción al pseudocódigo y sintaxis básica: Descripción corta

1. Propósito y beneficios
2. Reglas básicas de escritura

2. Estructuras de control en pseudocódigo: Descripción corta

1. Secuencias
2. Condicionales
3. Bucles

Actividades

- **Actividad 1: Escribir recetas en pseudocódigo** - Convertir una tarea cotidiana en pasos secuenciales en pseudocódigo y justificar las decisiones de flujo.
- **Actividad 2: Resuelve enigmas simples con condicionales** - Crear pseudocódigo que elija una acción en función de una percepción dada.
- **Actividad 3: Añadir bucles a un algoritmo** - Incorporar un bucle para repetir acciones hasta alcanzar una condición.

Evaluación

- Precisión y claridad de las estructuras de control en pseudocódigo (40%).
- Capacidad de convertir problemas en una secuencia lógica de pasos (30%).
- Uso correcto de bucles y condiciones en el pseudocódigo (30%).

Unidad 4: Unidad 4: Primer programa básico en Python: implementación de una decisión de agente

Objetivos de Aprendizaje

- Escribir y ejecutar un programa en Python con entradas simples y salidas claras.
- Utilizar estructuras if/else para decidir una acción ante percepciones básicas.
- Explicar, con palabras propias, cómo el agente decide la acción dada una percepción simple.

Contenidos Temáticos

1. Introducción a Python: sintaxis básica, variables y tipos simples: Descripción corta

1. Variables, tipos numéricos y cadenas
2. Operadores y expresiones simples

2. Entrada y salida en Python: Descripción corta

1. Uso de input()

2. Imprimir resultados con print()
3. Decisiones en Python: Descripción corta
 1. Estructuras if/else
 2. Evaluación de condiciones simples
4. Modelar percepciones y acciones de un agente: Descripción corta
 1. Percepciones como datos
 2. Acciones como respuestas del agente

Actividades

- **Actividad 1: Tu primer programa en Python** - Escribir un programa que pida una entrada y muestre una acción basada en una condición simple. Puntos clave: lectura de entrada, evaluación de condición, salida. Aprendizajes: uso básico de Python para decisiones simples.
- **Actividad 2: Decisiones con percepciones** - Construir un pequeño agente que, al recibir una percepción numérica, elija una acción diferente según rangos definidos.
- **Actividad 3: Explicación del flujo** - Describir oralmente o por escrito cómo la percepción conduce a una acción en el programa.

Evaluación

- Ejercicio práctico de programación con una decisión del agente (50%).
- Explicación conceptual de la lógica de decisión (30%).
- Lectura de código y detección de errores simples (20%).

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

Objetivos de Aprendizaje

- Crear estructuras if/else para decidir entre dos o más acciones.
- Usar bucles (for/while) para repetir tareas hasta cumplir una condición.
- Combinar condicionales y bucles para resolver problemas simples de IA básica.

Contenidos Temáticos

1. Conditional logic con if/else: Descripción corta
 1. Sintaxis básica
 2. Anidamiento y priorización
2. Bucles: for y while: Descripción corta
 1. Uso de contadores

2. Control de terminación

Actividades

- **Actividad 1: Control de acceso con if/else** - Crear un programa que permita o niegue el acceso según una contraseña simulada.
- **Actividad 2: Conteo de intentos con bucle** - Implementar un bucle que repita hasta obtener una condición exitosa y reporte el número de intentos.
- **Actividad 3: Juego de preguntas y respuestas** - Diseñar un cuestionario simple que use condicionales y bucles para evaluar respuestas.

Evaluación

- Evaluación de uso correcto de if/else (40%).
- Evaluación de implementación de bucles (30%).
- Integración de condicionales y bucles en una solución funcional (30%).

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

Objetivos de Aprendizaje

- Crear, acceder y modificar listas en Python.
- Almacenar percepciones en una lista y actualizarla según nuevas entradas.
- Almacenar acciones posibles y seleccionar una acción a partir de percepciones almacenadas.

Contenidos Temáticos

1. Listas en Python: Descripción corta
 1. Creación y acceso
 2. Modificación y eliminación
2. Iteración sobre listas: Descripción corta
 1. for y comprensión de listas
3. Memoria de percepciones y acciones en listas: Descripción corta
 1. Almacenamiento de percepciones
 2. Selección de acción basada en percepciones

Actividades

- **Actividad 1: Construyendo una memoria de percepciones** - Crear y manipular una lista de percepciones y mostrar cómo cambia con nuevas lecturas.

- **Actividad 2: Acciones disponibles en una lista** - Implementar una lista de acciones y seleccionar una acción según la última percepción.
- **Actividad 3: Recorrido e interpretación** - Recorrer percepciones almacenadas y proponer una acción por cada conjunto de percepciones.

Evaluación

- Uso correcto de listas para almacenar percepciones y acciones (40%).
- Manipulación de listas (30%).
- Demostración de selección de acción basada en percepciones (30%).

Unidad 7: Unidad 7: Depuración y resolución de errores en programas pequeños

Objetivos de Aprendizaje

- Identificar errores de sintaxis comunes en Python y su significado.
- Analizar errores de lógica y plantear soluciones correctivas.
- Aplicar estrategias de depuración: impresión de estados, pruebas simples, lectura de mensajes de error.

Contenidos Temáticos

1. Errores de sintaxis: Descripción corta
 1. Errores de escritura y puntuación
 2. Indicios del intérprete
2. Errores de lógica y pruebas: Descripción corta
 1. Razonamiento de flujo
 2. Casos de prueba simples
3. Depuración y herramientas básicas: Descripción corta
 1. Impresión de estados
 2. Uso de depuradores simples

Actividades

- **Actividad 1: Depuración de errores intencionados** - Corregir un programa con errores de sintaxis y líneas lógicas defectuosas.
- **Actividad 2: Pruebas simples** - Diseñar pruebas simples para validar entradas y salidas esperadas.
- **Actividad 3: Registro de depuración** - Añadir impresiones de estado para rastrear el comportamiento del programa.

Evaluación

- Capacidad para reconocer y corregir errores de sintaxis (40%).
- Detección y solución de fallos de lógica (40%).
- Aplicación de estrategias de depuración (20%).

Unidad 8: Unidad 8: IA en acción: conceptos y limitaciones de un agente básico

Objetivos de Aprendizaje

- Describir el ciclo de percepción-decisión-acción de un agente básico.
- Comparar un agente básico con enfoques de IA más avanzados (aprendizaje, planificación, incertidumbre).
- Ilustrar con un ejemplo sencillo cómo se evalúan percepciones y se elige una acción.

Contenidos Temáticos

1. Modelo de agente sencillo: percepciones, estado y acción: Descripción corta
 1. Entrada del entorno
 2. Procesamiento y decisión
2. Evaluación de percepciones y decisión de acción: Descripción corta
 1. Evaluación de condiciones
 2. Selección de acción
3. Limitaciones frente a IA avanzada: Descripción corta
 1. Ausencia de aprendizaje
 2. Limitaciones ante incertidumbre

Actividades

- **Actividad 1: Construcción de un mini-agente en Python** - Implementar percepciones simples y una acción basada en condiciones; discutir por qué es un agente básico.
- **Actividad 2: Discusión guiada** - Analizar qué capacidades faltan en este agente en comparación con IA avanzada (aprendizaje, planificación, razonamiento probabilístico).
- **Actividad 3: Diseño de experimento simple** - Proponer un experimento para comparar un agente básico con una versión más compleja (teóricamente) en un entorno simulado.

Evaluación

- Comprensión conceptual del ciclo percepción-acción (40%).
- Capacidad de comparar enfoques básicos frente a IA avanzada (30%).
- Demostración de un ejemplo simple de evaluación de percepciones y acción (30%).

