

Pensamiento computacional aplicado a la robótica educativa espacial

Ingeniería | Ingeniería de sistemas

Descripción del Curso

DESCRIPCIÓN

Esta unidad forma parte del curso Ingeniería de Sistemas y se centra en la implementación de programas para el control de un robot educativo en entornos de simulación de misiones espaciales. Se seleccionan lenguajes adecuados como Python o Scratch para favorecer la accesibilidad y la comprensión, especialmente en contextos educativos. El énfasis está en la legibilidad del código, la modularidad y la aplicación de buenas prácticas de codificación, con atención a pruebas y documentación para facilitar la colaboración, la repetibilidad y la transferencia de soluciones entre diferentes proyectos y grupos de estudiantes.

El curso busca que los estudiantes no solo aprendan a programar, sino que adopten una metodología de desarrollo orientada a proyectos: diseñar estructuras modulares, escribir código claro y mantenible, documentar decisiones de diseño y realizar pruebas para verificar el comportamiento del robot en simulaciones. Se promueve un aprendizaje colaborativo, donde la documentación y los recursos compartidos permiten reproducir resultados y ampliar las experiencias en futuros proyectos de robótica espacial educativa.

Dirigido a estudiantes a partir de 17 años, sin límite de edad superior, la unidad integra conceptos de programación, robótica educativa y simulación para fomentar habilidades prácticas aplicables a escenarios reales y educativos. Al finalizar la unidad, se espera que el alumno sea capaz de implementar soluciones modulares y bien documentadas que controlen robots simulados y verificar su correcto funcionamiento a través de pruebas estructuradas.

Competencias

COMPETENCIAS

- Competencia técnica: diseñar e implementar programas para controlar un robot educativo en simulaciones de misiones espaciales utilizando Python o Scratch, con énfasis en legibilidad y modularidad.
- Competencia analítica: aplicar principios de programación modular (funciones, módulos, separación de responsabilidades) para estructurar soluciones robustas y reutilizables.
- Competencia de calidad de software: escribir código claro, documentado y acompañado de pruebas que aseguren su correcto funcionamiento y faciliten la reutilización en contextos educativos.
- Competencia de comunicación y colaboración: documentar decisiones de diseño y trabajar en equipos para compartir recursos y reproducibilidad de experiencias de aprendizaje.

- Competencia de simulación y evaluación: diseñar simulaciones simples de misiones espaciales, ejecutar pruebas y analizar resultados para validar comportamientos deseados.

Requerimientos

REQUERIMIENTOS

- Conocimientos básicos de programación (variables, estructuras de control, bucles).
- Familiaridad con al menos un lenguaje de programación de alto nivel (Python o Scratch).
- Acceso a un entorno de desarrollo adecuado (un IDE para Python o la interfaz de Scratch) o plataformas de simulación disponibles.
- Capacidad para leer y producir documentación técnica y comentarios de código.
- Conexión a internet y herramientas de colaboración para intercambio de materiales y versiones (opcional pero recomendable).
- Compromiso con buenas prácticas de pruebas y documentación para facilitar la repetibilidad.

Unidades del Curso

Unidad 1: Unidad 1: Componentes y pensamiento computacional en robótica educativa espacial

Objetivos de Aprendizaje

- Identificar y describir los sensores, actuadores y controladores de un robot educativo espacial, y explicar su papel en una tarea típica de misión educativa.
- Aplicar conceptos de descomposición y abstracción para entender cómo un robot percibe el entorno y ejecuta acciones básicas.
- Reconocer patrones simples en datos sensoriales y diseñar algoritmos básicos que coordinan movimiento y recopilación de datos.

Contenidos Temáticos

1. **Componentes de un robot educativo espacial** – Descripción de sensores (distancia, color, acelerómetro), actuadores (motores, servomotores) y controladores (microcontroladores, MCU/SoC).
2. **Interacción entre componentes y pensamiento computacional** – Cómo se traducen percepciones en acciones mediante descomposición, abstracción y diseño de algoritmos.
3. **Introducción a proyectos educativos de robótica espacial** – Ejemplos de tareas simples y simulaciones para practicar pensamiento computacional.

Actividades

- **Actividad 1: Exploración de un robot educativo espacial** – Los estudiantes identifican sensores, actuadores y controladores en un robot o simulación. Breve descripción de la función de cada componente, con un diagrama simple. Puntos clave: entender qué hace cada parte, relaciones entre sensores y actuadores, y cómo un controlador orquesta acciones.
- **Actividad 2: Descomposición de una tarea simple** – Descomponer una tarea de misión en pasos computacionales básicos (capturar datos, procesar, decidir, actuar). Puntos clave: dividir en subtareas manejables y asignar responsabilidades a cada componente.
- **Actividad 3: Abstracción y diseño de un algoritmo básico** – Construcción de un algoritmo simple para hacer que el robot se desplace a una posición objetivo utilizando un conjunto mínimo de sensores. Puntos clave: identificar entradas, salidas, condiciones y pasos críticos.

Evaluación

- Competencia 1: Identificación de componentes y funciones del robot (tipo de sensor, actuador, controlador). Evaluación a través de un cuestionario corto y un diagrama de flujo sencillo.
- Competencia 2: Descomposición y abstracción aplicadas a una tarea básica. Evaluación mediante un entregable que describa la descomposición de una tarea y la abstracción de sus datos clave.
- Competencia 3: Diseño de un algoritmo básico para movimiento. Evaluación por un pseudocódigo o diagrama de actividad y su explicación de cómo coordina sensores y actuadores.

Unidad 2: Unidad 2: Pensamiento computacional y resolución de problemas en misiones espaciales

Objetivos de Aprendizaje

- Relacionar la resolución de problemas de misión con las etapas de pensamiento computacional: descomposición, abstracción, reconocimiento de patrones y diseño de algoritmos.
- Analizar casos de estudio de misiones espaciales (simuladas) donde el pensamiento computacional facilita decisiones y acciones del robot.
- Comparar enfoques de resolución de problemas y justificar la elección de estrategias en escenarios educativos.

Contenidos Temáticos

1. **Introducción a misiones espaciales simuladas** – Características, objetivos y límites de las simulaciones en educación.
2. **Pensamiento computacional aplicado a la resolución de problemas** – Cómo descomposición, abstracción y reconocimiento de patrones orientan la toma de decisiones.
3. **Casos de estudio educativos** – Ejemplos de misiones y soluciones implementadas por estudiantes con robótica educativa.

Actividades

- **Actividad 1: Análisis de un caso de misión simulada** – Lectura de un caso y extracción de los pasos computacionales necesarios para completar la misión. Puntos clave: comprender requisitos, identificar entradas y salidas, y justificar elecciones de enfoque.
- **Actividad 2: Taller de resolución de problemas** – En pequeños grupos, plantear una situación de misión, descomponerla, abstraer variables relevantes y proponer soluciones computacionales. Puntos clave: colaboración, claridad en las decisiones y límites asumidos.
- **Actividad 3: Presentación de soluciones** – Compartir soluciones propuestas, discutir ventajas y limitaciones, y recibir retroalimentación para mejorar la correspondencia entre problema y solución.

Evaluación

- Competencia 1: Capacidad para descomponer un problema de misión y mapearlo a componentes computacionales. Evaluación mediante un informe analítico de caso.
- Competencia 2: Uso correcto de abstracción y reconocimiento de patrones para seleccionar variables relevantes. Evaluación mediante una matriz de decisión y justificación.
- Competencia 3: Calidad de la solución propuesta y su claridad de comunicación. Evaluación mediante presentación y feedback de pares.

Unidad 3: Unidad 3: Planificación de tareas robóticas en escenarios espaciales (descomposición, abstracción y reconocimiento de patrones)

Objetivos de Aprendizaje

- Descomponer tareas complejas en subtareas manejables para su implementación en un robot educativo espacial.
- Utilizar abstracción para crear modelos simples de escenarios espaciales y de sus restricciones.
- Reconocer patrones en datos sensoriales para anticipar estados del entorno y adaptar la planificación.

Contenidos Temáticos

1. **Descomposición de tareas espaciales** – Métodos para dividir objetivos en acciones secuenciadas y paralelas.
2. **Abstracción y modelado de escenarios** – Representación simplificada de variables clave (posiciones, velocidades, obstáculos).
3. **Reconocimiento de patrones para la planificación** – Identificación de secuencias recurrentes y señales relevantes en datos sensoriales.
4. **Límites y supuestos en la planificación** – Consideraciones prácticas, de hardware y de seguridad en misiones educativas.

Actividades

- **Actividad 1: Descomposición de una tarea de exploración** – Dividir una misión de exploración en subtareas, asignando responsables a sensores y actuadores. Puntos clave: secuencias, dependencias y tiempos.
- **Actividad 2: Modelado abstracto de un escenario espacial** – Crear un modelo simplificado (variables clave, restricciones) para facilitar la planificación.
- **Actividad 3: Detección de patrones y planificación adaptativa** – Analizar datos de sensores para identificar patrones y ajustar la secuencia de acciones ante cambios en el entorno. Puntos clave: robustez y flexibilidad.
- **Actividad 4: Evaluación de límites y supuestos** – Identificar límites de la planificación y discutir cómo validarlos en simulación.

Evaluación

- Competencia 1: Capacidad de dividir una tarea compleja en subtareas y asignar dependencias temporales. Evaluación mediante un diagrama de tareas y justificación.
- Competencia 2: Modelado abstracto correcto del escenario de misión y adecuación del modelo a la planificación. Evaluación mediante un informe de modelado.
- Competencia 3: Reconocimiento de patrones y adaptación de la planificación en respuesta a cambios. Evaluación mediante simulación o estudio de caso y discusión.

Unidad 4: Unidad 4: Diseño de algoritmos para control de movimiento, toma de decisiones y procesamiento de datos sensoriales

Objetivos de Aprendizaje

- Desarrollar algoritmos de movimiento básicos (por ejemplo, navegación y colisión) para un robot en entornos simulados.
- Crear reglas de decisión simples para la toma de decisiones autónoma del robot ante diferentes escenarios.
- Diseñar procesos de procesamiento básico de datos sensoriales (filtrado, umbrales, detección de eventos).

Contenidos Temáticos

1. **Algoritmos de control de movimiento** – Principios básicos de navegación, límites y seguridad.
2. **Toma de decisiones automatizada** – Reglas simples, estados y transiciones.
3. **Procesamiento de datos sensoriales** – Filtrado, umbrales, detección de eventos y representación de datos.

Actividades

- **Actividad 1: Diseño de un algoritmo de movimiento** – Crear un flujo de control para navegar hasta un objetivo manteniendo distancia de seguridad. Puntos clave: lógica de control, robustez ante obstáculos simples.
- **Actividad 2: Implementación de reglas de decisión** – Definir un conjunto de condiciones para que el robot elija entre acciones como avanzar, detenerse o cambiar de ruta. Puntos clave: claridad de condiciones y acciones

resultantes.

- **Actividad 3: Procesamiento de datos sensoriales** – Aplicar técnicas simples de filtrado y umbrales para detectar eventos (p. ej., proximidad a obstáculo). Puntos clave: interpretación de señales y robustez frente a ruido.

Evaluación

- Competencia 1: Funcionamiento correcto de un algoritmo de movimiento en simulación y su explicación de elección de parámetros.
- Competencia 2: Validez de las reglas de decisión ante distintos escenarios simulados.
- Competencia 3: Calidad del procesamiento de datos sensoriales y su impacto en la toma de decisiones.

Unidad 5: Unidad 5: Implementación de programas en Python o Scratch para control de robots espaciales y prácticas de legibilidad y modularidad

Objetivos de Aprendizaje

- Aplicar principios de programación modular: funciones, módulos y separación de responsabilidades.
- Escribir código claro y documentado que sea fácil de mantener y ampliar en proyectos educativos de robótica espacial.
- Desarrollar simulaciones simples de misiones espaciales y realizar pruebas para verificar su comportamiento.

Contenidos Temáticos

1. **Lenguajes de programación para robótica educativa** – Python y Scratch, ventajas y casos de uso.
2. **Estructuras y modularidad** – Funciones, clases básicas y organización de proyectos.
3. **Pruebas, depuración y documentación** – Estrategias para validar código y mantener registros de aprendizaje.

Actividades

- **Actividad 1: Programación modular en Python** – Implementar un controlador básico para movimiento, con funciones independientes para lectura de sensores, toma de decisiones y acción. Puntos clave: reutilización, claridad y pruebas unitarias.
- **Actividad 2: Proyecto en Scratch** – Construir un programa de misión espacial en Scratch que simule navegación y toma de decisiones. Puntos clave: flujo lógico, sprites y eventos.
- **Actividad 3: Prácticas de legibilidad y documentación** – Añadir comentarios, docstrings o notas en Scratch, y crear un README con instrucciones de uso y pruebas realizadas. Puntos clave: mantenibilidad y colaboración.

Evaluación

- Competencia 1: Implementación de un programa modular con funciones bien definidas y pruebas básicas. Evaluación mediante revisión de código y ejecución en simulación.

- Competencia 2: Calidad de la documentación y legibilidad del código. Evaluación mediante checklist de buenas prácticas y comentarios explicativos.
- Competencia 3: Capacidad de realizar pruebas de simulación y validar resultados. Evaluación mediante informe de pruebas y resultados observables.