**C.E. en IA y BD – PROGRAMACIÓN DE INTELIGENCIA ARTIFICIAL**

**Nombre y Apellidos:**

**ACTIVIDAD 03**

**INTRODUCCIÓN**

**¡Bienvenido al equipo de desarrollo de KITT, el coche autónomo más avanzado del mundo! En este proyecto, serás el encargado de programar las funcionalidades básicas que permitirán a KITT interactuar inteligentemente con su entorno y con el conductor.**

**Ten en cuenta que estamos realizando esta actividad para prácticas las instrucciones y métodos vistos en clase, por lo que puede no ser la forma óptima de conseguir la misma funcionalidad.**

**LIMITACIONES**

**En cada “Parte”, tendrás limitadas las instrucciones y métodos que puedes utilizar, evitando por completo métodos que no hayamos visto en clase. Las instrucciones y métodos expuestos en los requisitos tendrán que utilizarse al menos una vez.**

**PARTE 1: INICIO**

1. **MISIÓN**

**En esta primera parte, configurarás las variables iniciales necesarias para la simulación del viaje de KITT. El objetivo es solicitar al usuario información esencial y establecer parámetros clave para el viaje.**

1. **CONCEPTOS REQUERIDOS**

* **Condicional ternario**
* **Estructura match**
* **Operadores aritméticos y lógicos**
* **Funciones input y print**
* **Variables y estructuras de datos**
* **Pasarlo bien**

1. **INSTRUCCIONES**

* **Mensaje de bienvenida a “Kitt, tu coche autónomo”.**
* **Solicitar la distancia total y el nivel de combustible al usuario.**
  + **Declarar dos variables que recojan la distancia total del viaje (suponemos que números enteros correspondientes a los km) y el nivel del combustible actual (suponemos números enteros correspondientes a los litros).**
  + **Declarar una variable que guarde la distancia recorrida (inicializar en 0).**
* **Selección del modo de conducción.**
  + **Declara una variable que recoja el modo de conducción entre “Eco, Sport y Normal”.**
  + **Transforma el string introducido por el usuario a mayúsculas, para después utilizar la estructura match y evitar errores (.capitalize()).**
  + **Declara una variable que recoja el consumo por litros cada 10km, el valor de esta variable será dado en la estructura match siguiente.**
  + **Estructura match: dependiendo del modo seleccionado por el usuario, el consumo por 10km será:**
    - **Eco: 2l/10km**
    - **Sport: 6l/10km**
    - **Normal: 3l/10km**
    - **Resto de casos: 3l/10km**
* **Realiza una comprobación inicial del combustible:**
  + **Utiliza un condicional ternario, el cual mostrará “No hay combustible suficiente” si:**
  + **En caso contrario, mostrará “Hay combustible suficiente para realizar el viaje, ¡Adelante!”.**

1. **EJEMPLO DE RESULTADO EN CONSOLA**

Texto

Descripción generada automáticamente

**PARTE 2: VIAJE ANTES QUE DESTINO**

1. **MISIÓN**

**En esta segunda parte, vamos a simular que ciertos eventos pueden ocurrir en la carretera, haciendo consumir combustible adicional al vehículo en la ruta.**

1. **CONCEPTOS REQUERIDOS**

* **Condicional if**
* **Bucle while**
* **Bucle for**
* **enumerate**
* **Break**
* **Random 🡪 Importar librería**
* **Round**

1. **INSTRUCCIONES**

* **Define una lista de eventos en ruta que contenga los siguientes eventos:**
  + **Tráfico denso**
  + **Obras en la carretera**
  + **Clima adverso**
* **Configura un bucle while que se ejecuta mientras la distancia recorrida sea menor que la distancia total. Anidado dentro del bucle:**
  + **Incrementa la distancia recorrida en 10km para simular el avance del vehículo.**
  + **Calcula y resta del nivel de combustible el consumo correspondiente por cada 10km basado en el consumo por 10km definido en la Parte I.**
  + **Muestra la distancia recorrida y el combustible restante.**
  + **Después de actualizar el nivel de combustible, verifica con un condicional de tipo if si es menor o igual a cero y en caso afirmativo muestra el mensaje “Te has quedado sin combustible”. Utiliza un break para terminar con el bucle y por tanto con el programa.**
  + **Anida un bucle for para iterar sobre la lista de eventos en ruta haciendo uso de enumerate(). Para cada evento:**
    - **Genera un número aleatorio entre 0 y 1 utilizando random().**
    - **Redonda el número aleatorio utilizando round para obtener o bien 0 o bien 1.**
    - **Resta el valor al nivel de combustible.**
    - **Muestra un mensaje “El evento [evento] te ha dejado con [nivel de combustible] litros de combustible”.**
* **Después del bucle while, verifica si la distancia recorrida es mayor o igual a la distancia total y si el nivel de combustible es mayor que cero.**
  + **Si es así muestra un mensaje de felicitación mostrando la distancia recorrida y el nivel de combustible restante.**

1. **EJEMPLO DE RESULTADO EN CONSOLA**

Texto

Descripción generada automáticamente

**PARTE 3: DESTINO**

1. **MISIÓN**

**Esta última parte, va a permitir al usuario tomar decisiones dependiendo de los eventos ocurridos. Para realizar esta parte, necesitarás borrar el bloque de código que se encuentra dentro del for que recorre los eventos en ruta y sustituirlo por el nuevo bloque de código que se explicará a continuación. Puedes utilizar un fichero de Python diferente.**

1. **CONCEPTOS REQUERIDOS**

* **Condicional if elif else**
* **Continue**
* **Bucle for**
* **enumerate**
* **Break**
* **Random 🡪 Importar librería**
* **Round**

1. **INSTRUCCIONES**

* **Sustituye el contenido del bucle for por el código que escribirás a continuación. Deja únicamente la generación del número aleatorio entre 0 y 1 (sin redondear).**
* **Para cada iteración del bucle for, es decir, cada evento de la lista de eventos en ruta, generarás un número aleatorio entre 0 y 1.**
* **Anida un condicional en el bucle for. Si el número aleatorio generado es < 0.9, mostrará un mensaje de “No ocurre nada” y hará uso de la instrucción continue para pasar al siguiente evento.**
* **Si el número es > 0.9, se tendrá que gestionar el evento en un else.**
  + **Dentro de este else, muestra un mensaje de “¡Ha ocurrido un evento!” y seguido el evento ocurrido obtenido de la lista.**
  + **Anida en el else un condicional que incluya decisiones para cada evento de la lista.**
  + **Por ejemplo:**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. **EJEMPLO DE RESULTADO EN CONSOLA**

Texto

Descripción generada automáticamente