

**CASOS DE USO**

Tabla de contenidos

|  |  |  |
| --- | --- | --- |
| 1 | **La plataforma debe ser capaz de moverse de manera biomimética, imitando patrones de locomoción de cuadrúpedos en la naturaleza.** | …1 |
| 2 | **La plataforma debe poder navegar y maniobrar de manera efectiva en entornos complejos, como terrenos irregulares, obstáculos y condiciones cambiantes.** | …2 |
| 3 | **Debe contar con una interfaz de usuario intuitiva y efectiva que permita la interacción humano-máquina de forma segura y comprensible.** | …3 |
| 4 | **La plataforma debe integrar sistemas de control autónomo que le permitan tomar decisiones adaptativas en tiempo real basadas en la retroalimentación sensorial y las condiciones del entorno.** | …4 |
| 5 | **Debe ser capaz de adaptarse a situaciones imprevistas o escenarios no programados, demostrando flexibilidad y toma de decisiones autónomas.** | …5 |
| 6 | **Se requiere que la plataforma opere de manera eficiente desde el punto de vista energético para maximizar su autonomía y minimizar el impacto ambiental.** | …6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1. La plataforma debe ser capaz de moverse de manera biomimética, imitando patrones de locomoción de cuadrúpedos en la naturaleza.** | | | |  |
|  | **Universidad Autónoma de Occidente**  **DESARROLLO DE UNA PLATAFORMA POLIVALENTE BIOINSPIRADA EN UN CUADRÚPEDO PARA INTERACCIÓN HUMANO-MAQUINA EN ENTORNOS COMPLEJOS** | | Rev.:  000 | |
| Title:  **CASO DE USO**  La plataforma debe ser capaz de moverse de manera biomimética, imitando patrones de locomoción de cuadrúpedos en la naturaleza. | | Document:  CUR-001 | Page:  1 de 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Historic review | | | |
| Rev. | Change description | Author | Date |
| 001 | Document construction |  |  |
| 002 | Structural Change |  |  |

|  |  |
| --- | --- |
| **GENERAL INFORMATION** |  |
| **Actors** | Usuario, Jetson Orion, Raspberry Pi, Esp32 |
| **Propuse** | Asegurar que la plataforma pueda replicar los movimientos naturales de cuadrúpedos para mejorar su eficiencia y adaptabilidad en diversos entornos. |
| **Summarize** | describe cómo el usuario puede iniciar y controlar el movimiento biomimético de la plataforma. |
| **Type** |  |

|  |  |
| --- | --- |
| **Curso normal de los eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| Actor (Usuario): Envía un comando de inicio de locomoción a través de la interfaz en la Jetson Orion. | La Jetson Orion inicia los procesos de percepción y análisis del entorno en respuesta al comando de inicio. |
| Actor (Jetson Orion): Procesa datos de percepción de sensores (lidar, cámara, micrófonos) para analizar el entorno y determinar la ruta óptima. Calcula el patrón de locomoción biomimético y envía el esquema de movimiento a la Raspberry Pi. | La Jetson Orion utiliza datos de sensores para identificar obstáculos y terreno, generando un esquema de movimiento biomimético y un comando de desplazamiento para la Raspberry Pi. |
| Actor (ESP32 Sensorial): Envía datos en tiempo real de encoders, IMU y otros sensores a la Jetson Orion, permitiéndole ajustar el patrón de locomoción. | La Jetson Orion ajusta dinámicamente el patrón de locomoción con la información proporcionada por el ESP32 Sensorial (IMU, encoders, etc.). |
| Actor (Raspberry Pi): Recibe el esquema de movimiento de la Orion y distribuye los comandos de locomoción a cada ESP32 de las patas.  Actor (ESP32 de cada pata): Recibe órdenes, calcula la cinemática inversa para los tres motores brushless y determina los ángulos exactos para el movimiento de cada motor. | La Raspberry Pi interpreta el esquema de movimiento y distribuye órdenes mediante CAN a los ESP32, los cuales ejecutan cálculos de cinemática inversa para establecer ángulos específicos para los motores. |
| Actor (ESP32 de cada pata): Ejecuta el movimiento de las patas de forma sincronizada siguiendo el patrón biomimético.  Actor (Raspberry Pi): Supervisa y ajusta la posición de las patas en tiempo real. | Cada ESP32 de las patas sincroniza el movimiento de los motores brushless, proporcionando un desplazamiento fluido y estable. La Raspberry Pi asegura la precisión en la posición de cada pata. |
| Actor (Jetson Orion): Recibe datos de posición y orientación del robot desde el ESP32 Sensorial, permitiendo ajustes en trayectoria o velocidad si es necesario. | La Jetson Orion monitoriza y adapta la trayectoria del robot en función de los datos en tiempo real del entorno y el estado del robot. |
| Actor (Raspberry Pi): Envía un mensaje de finalización a la Jetson Orion cuando el destino se ha alcanzado o se completa la tarea. | La Jetson Orion recibe la señal de finalización de la Raspberry Pi, indicando que la tarea ha sido completada exitosamente. |

|  |
| --- |
| **Curso alternativo de los eventos** |
| **Respuesta del sistema** |
| Error en la percepción o sensores   * La Jetson Orion detecta una falta de respuesta o un error en los datos de sensores (lidar, cámara, IMU, encoders, etc.). Notifica al usuario mediante un mensaje de error en la interfaz y detiene la locomoción hasta que se restablezca la comunicación con los sensores.   Fallo en el cálculo de la cinemática inversa   * Uno de los ESP32 de las patas no puede calcular la cinemática inversa por un error en los datos de entrada o una desconexión con la Raspberry Pi. La Raspberry Pi recibe el error y envía una notificación a la Jetson Orion, que pausa el movimiento del robot y solicita una recalibración o diagnóstico.   Desincronización entre patas   * La Raspberry Pi detecta una asimetría en el movimiento de las patas (por ejemplo, retraso en una pata). Ajusta los comandos de CAN para reestablecer la sincronización. Si persiste el problema, notifica a la Jetson Orion, que detiene la locomoción y muestra un mensaje de diagnóstico al usuario.   Obstáculo imprevisto en la trayectoria   * La Jetson Orion identifica un obstáculo no previsto mediante el lidar o cámara y recalcula la ruta de manera automática para evitarlo, generando un nuevo esquema de movimiento que envía a la Raspberry Pi. Si no se puede evitar el obstáculo, la locomoción se detiene, y se notifica al usuario.   Interrupción en la comunicación CAN   * La Raspberry Pi detecta un fallo en la comunicación CAN con uno o varios ESP32 de las patas. Intenta reestablecer la conexión. Si no es posible, la Jetson Orion recibe una alerta de error y muestra un mensaje de emergencia, sugiriendo revisar las conexiones de las patas afectadas.   Error en el control de campo magnético de los motores   * Si un driver de motor brushless reporta un fallo en el control de campo magnético (por ejemplo, sobrecalentamiento o falta de corriente), el ESP32 correspondiente informa a la Raspberry Pi, que detiene la operación de esa pata. La Jetson Orion recibe el informe y ajusta el movimiento para compensar la pata afectada o sugiere una reparación.   Fallo en el sistema de monitoreo en tiempo real   * Si la Jetson Orion no recibe datos en tiempo real del ESP32 Sensorial (por ejemplo, pérdida de datos de encoders o IMU), alerta al usuario y realiza una pausa preventiva. Una vez restablecida la comunicación, se reanuda el monitoreo y se procede a la evaluación de estado antes de continuar.   Finalización de la locomoción antes de tiempo   * Si la Jetson Orion detecta condiciones inseguras o un comando de cancelación del usuario, detiene la locomoción inmediatamente. La Raspberry Pi y los ESP32 de las patas reciben el comando de parada segura, y el sistema finaliza el movimiento sin errores críticos. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2. La plataforma debe poder navegar y maniobrar de manera efectiva en entornos complejos, como terrenos irregulares, obstáculos y condiciones cambiantes.** | | | |  |
| Imagen que contiene Logotipo  Descripción generada automáticamente | **Universidad Autónoma de Occidente**  **DESARROLLO DE UNA PLATAFORMA POLIVALENTE BIOINSPIRADA EN UN CUADRÚPEDO PARA INTERACCIÓN HUMANO-MAQUINA EN ENTORNOS COMPLEJOS** | | Rev.:  000 | |
| Title:  **CASO DE USO**  La plataforma debe poder navegar y maniobrar de manera efectiva en entornos complejos, como terrenos irregulares, obstáculos y condiciones cambiantes. | | Document:  CUR-001 | Page:  1 de 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Historic review | | | |
| Rev. | Change description | Author | Date |
| 001 | Document construction |  |  |
| 002 | Structural Change |  |  |

|  |  |
| --- | --- |
| **GENERAL INFORMATION** |  |
| **Actors** | Usuario, Jetson Orion, Raspberry Pi, Esp32 |
| **Propuse** | Garantizar que la plataforma pueda desplazarse eficientemente en diversos terrenos y condiciones. |
| **Summarize** | Este caso de uso describe cómo el usuario puede iniciar la navegación autónoma y cómo la plataforma maneja entornos complejos. |
| **Type** |  |

|  |  |
| --- | --- |
| **Curso normal de los eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| **Actor (Usuario):** Envía un comando de inicio de navegación a través de la interfaz de usuario en la Jetson Orion. | La **Jetson Orion** activa todos los sistemas de percepción y análisis del entorno en respuesta al comando de inicio. |
| **Actor (Jetson Orion)**: Procesa los datos de percepción de sensores (lidar, cámara, etc.) para identificar obstáculos, condiciones del terreno y posibles rutas.  **Actor (ESP32 Sensorial)**: Envía continuamente datos de los encoders y del sensor IMU para que la Orion evalúe el estado del robot y del terreno. | La **Jetson Orion** procesa la información de los sensores y del ESP32 Sensorial para identificar obstáculos, el tipo de terreno y las mejores rutas disponibles. |
| **Actor (Jetson Orion):** Calcula la maniobra adecuada (esquema de movimiento) para sortear los obstáculos y adaptarse a la irregularidad del terreno. Envía las instrucciones de movimiento a la Raspberry Pi. | La **Jetson Orion** genera un esquema de maniobras adaptativas (movimiento de patas, ajustes de velocidad, etc.) y envía las instrucciones de locomoción a la Raspberry Pi. |
| **Actor (ESP32 Sensorial):** Sigue proporcionando datos en tiempo real sobre inclinación y estabilidad del terreno a la Jetson Orion.  **Actor (Jetson Orion):** Ajusta la maniobra y envía comandos correctivos a la Raspberry Pi en caso de detectar cambios en el terreno o nuevos obstáculos. | El **ESP32 Sensorial** monitorea inclinación y estabilidad, enviando información continua a la Jetson Orion. La Jetson Orion responde con comandos correctivos si detecta cambios en el terreno. |
| **Actor (Jetson Orion):** Monitorea continuamente la posición del robot mediante los datos del ESP32 Sensorial y realiza ajustes de trayectoria si es necesario. | La **Jetson Orion** utiliza los datos del ESP32 Sensorial para ajustar la trayectoria en tiempo real, manteniendo al robot en la ruta deseada incluso con irregularidades en el terreno. |
| **Actor (Usuario):** Envía un comando para finalizar la navegación o el robot alcanza el destino.  **Actor (Jetson Orion):** Envía la orden de detener el movimiento a la Raspberry Pi, que distribuye el comando a los ESP32 de las patas para detenerse de forma segura. | La **Jetson Orion** detiene la navegación al recibir el comando del usuario o al alcanzar el destino, enviando un comando de parada segura a la Raspberry Pi, que se transmite a cada ESP32 de las patas para detenerse gradualmente. |

|  |
| --- |
| **Curso alternativo de los eventos** |
| **Respuesta del sistema** |
| **Error en Percepción del Entorno**   * La Jetson Orion detecta una falta de respuesta o datos inconsistentes de los sensores (lidar, cámara, etc.). Notifica al usuario a través de la interfaz y detiene la navegación para prevenir riesgos hasta que los sensores vuelvan a responder adecuadamente.   **Inestabilidad Detectada en el Terreno**   * Los datos del ESP32 Sensorial indican inestabilidad o inclinación extrema en el terreno. La Jetson Orion recalcula la maniobra para adoptar una postura de estabilidad y envía un comando de ajuste a la Raspberry Pi, solicitando movimientos de patas que estabilicen al robot en tiempo real.   **Desincronización entre Patas**   * La Raspberry Pi detecta una desincronización en los comandos de las patas (por ejemplo, retraso en la respuesta de un ESP32 de una pata). Intenta reestablecer la sincronización enviando una nueva señal a los ESP32 afectados. Si el problema persiste, la Jetson Orion recibe una notificación y pausa la navegación, solicitando una revisión.   **Obstáculo Imposible de Sortear**   * La Jetson Orion identifica un obstáculo insalvable mediante el lidar o la cámara. Intenta recalcular la ruta, pero si no encuentra una alternativa, notifica al usuario y detiene la locomoción.   **Interrupción en la Comunicación CAN**   * La Raspberry Pi pierde comunicación CAN con uno o más ESP32 de las patas. Intenta reestablecer la conexión, y si no tiene éxito, la Jetson Orion recibe una alerta de error y muestra un mensaje de emergencia al usuario, sugiriendo revisar las conexiones de las patas afectadas.   **Error en Control de Campo Magnético en los Motores**   * Uno de los drivers de motor brushless reporta un fallo en el control de campo magnético (por ejemplo, sobrecalentamiento). El ESP32 de la pata afectada informa a la Raspberry Pi, que ajusta la maniobra general para compensar la pata afectada. La Jetson Orion también recibe el reporte y notifica al usuario con la sugerencia de revisar el motor.   **Fallo en el Monitoreo en Tiempo Real**   * La Jetson Orion detecta una falta de datos en tiempo real del ESP32 Sensorial (por ejemplo, pérdida de datos de encoders o IMU). En respuesta, la Jetson Orion detiene la navegación preventivamente y alerta al usuario, recomendando una revisión del sensor.   **Finalización de la Navegación antes de lo Planeado**   * La Jetson Orion recibe un comando de cancelación del usuario o detecta condiciones inseguras que requieren detener la operación. Detiene la navegación inmediatamente y envía un comando de parada segura a la Raspberry Pi, que se comunica con cada ESP32 de las patas para finalizar el movimiento sin errores críticos. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3. Debe contar con una interfaz de usuario intuitiva y efectiva que permita la interacción humano-máquina de forma segura y comprensible.** | | | |  |
| Imagen que contiene Logotipo  Descripción generada automáticamente | **Universidad Autónoma de Occidente**  **DESARROLLO DE UNA PLATAFORMA POLIVALENTE BIOINSPIRADA EN UN CUADRÚPEDO PARA INTERACCIÓN HUMANO-MAQUINA EN ENTORNOS COMPLEJOS** | | Rev.:  000 | |
| Title:  **CASO DE USO**  Debe contar con una interfaz de usuario intuitiva y efectiva que permita la interacción humano-máquina de forma segura y comprensible. | | Document:  CUR-001 | Page:  1 de 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Historic review | | | |
| Rev. | Change description | Author | Date |
| 001 | Document construction |  |  |
| 002 | Structural Change |  |  |

|  |  |
| --- | --- |
| **GENERAL INFORMATION** |  |
| **Actors** | Usuario, Jetson Orion, Raspberry Pi, Esp32 |
| **Propuse** | Facilitar la interacción del usuario con la plataforma de manera segura y eficiente. |
| **Summarize** | Este caso de uso describe cómo el usuario interactúa con la interfaz para controlar y monitorear la plataforma. |
| **Type** |  |

|  |  |
| --- | --- |
| **Curso normal de los eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| **Actor (Usuario)**: Accede a la interfaz de usuario en la pantalla conectada a la Jetson Orion para iniciar la interacción con el sistema. | La **Jetson Orion** inicializa la interfaz de usuario mostrando el estado general del sistema y opciones de operación, proporcionando al usuario un punto de inicio claro y comprensible para la interacción. |
| **Actor (Jetson Orion)**: Muestra en la interfaz de usuario información en tiempo real sobre el estado del sistema, como la carga de la batería, estado de los motores, y datos de sensores relevantes (por ejemplo, datos del lidar y de la IMU). | La **Jetson Orion** procesa y muestra en la interfaz los datos relevantes del sistema, ofreciendo al usuario información completa sobre la energía, estado de sensores, y otros elementos importantes para la toma de decisiones. |
| **Actor (Usuario):** Utiliza la interfaz para seleccionar un modo de operación (por ejemplo, modo de ahorro de energía, modo de navegación, modo de diagnóstico). | La **Jetson Orion** recibe la selección del usuario, y envía el comando a la Raspberry Pi, que ajusta los parámetros de operación de los ESP32 de las patas según el modo seleccionado. |
| **Actor (Jetson Orion):** Procesa la selección del usuario y envía el comando de operación a la Raspberry Pi para ejecutar el modo seleccionado.  **Actor (Raspberry Pi):** Interpreta el comando y ajusta la configuración de los ESP32 de las patas según el modo de operación solicitado. | La **Raspberry Pi** reconfigura el comportamiento de los motores y otros componentes según el modo solicitado, mientras la **Jetson Orion** actualiza la interfaz para indicar que el sistema está funcionando en el modo seleccionado. |
| **Actor (ESP32 Sensorial):** Envía datos continuos de la IMU, encoders y otros sensores a la Jetson Orion.  **Actor (Jetson Orion):** Actualiza la interfaz de usuario con información en tiempo real sobre el progreso y estado del sistema, asegurando que el usuario reciba una retroalimentación clara y constante**.** | La **Jetson Orion** recibe datos del ESP32 Sensorial y presenta un flujo de información en tiempo real en la interfaz, permitiendo que el usuario observe el progreso y estado de las operaciones en curso. |
| **Actor (Jetson Orion):** Detecta situaciones de estado crítico (batería baja, sobrecalentamiento de motores, errores de sensor) y muestra mensajes de alerta en la interfaz de usuario. Permite que el usuario tome decisiones para detener el sistema o ajustar la operación. | La **Jetson Orion** evalúa los datos del sistema en busca de condiciones críticas y, en caso de detectarlas, alerta al usuario de manera inmediata a través de mensajes en la interfaz para que pueda tomar decisiones de manera informada. |
| **Actor (Usuario):** Selecciona la opción de finalizar la operación desde la interfaz.  **Actor (Jetson Orion):** Envía un comando de cierre a la Raspberry Pi para detener de forma segura la operación de los motores y desconectar sensores no críticos.  **Actor (Raspberry Pi):** Detiene la locomoción y confirma la desactivación de los motores, enviando una señal de confirmación a la Jetson Orion. | La **Jetson Orion** envía un comando de apagado controlado a la **Raspberry Pi**, que asegura el apagado seguro de los motores y sensores no esenciales. La interfaz de usuario muestra un mensaje de confirmación de cierre seguro, informando al usuario que la operación ha finalizado correctamente. |

|  |
| --- |
| **Curso alternativo de los eventos** |
| **Respuesta del sistema** |
| **Fallo en la Visualización de Estado en la Interfaz**   * La **Jetson Orion** detecta un error en la comunicación con uno o más sensores (lidar, IMU, encoders) y muestra un mensaje en la interfaz indicando que la información de estado podría estar incompleta. Informa al usuario del problema y sugiere una revisión o un reinicio de la conexión de sensores.   **Selección de Modo de Operación No Disponible**   * La **Jetson Orion** recibe un comando del usuario para activar un modo que requiere más recursos de los disponibles (por ejemplo, un modo de alta potencia con batería baja). Notifica al usuario en la interfaz de que el modo seleccionado no está disponible por limitaciones de energía y sugiere opciones alternativas.   **Error en el Procesamiento de Comandos del Usuario**   * La **Jetson Orion** no puede procesar un comando específico debido a un fallo en el sistema o un error en la comunicación con la Raspberry Pi. Muestra un mensaje en la interfaz explicando que el comando no pudo ejecutarse y solicita al usuario que intente nuevamente o seleccione otro comando.   **Interrupción de Datos en Tiempo Real desde Sensores**   * La **Jetson Orion** detecta una interrupción en los datos en tiempo real del ESP32 Sensorial (por ejemplo, falta de datos de IMU o encoders). La interfaz muestra una alerta al usuario indicando que la información de posición y estado puede no estar actualizada y recomienda suspender la operación si el problema persiste.   **Detección de Situación Crítica sin Respuesta del Usuario**   * La **Jetson Orion** detecta una situación crítica (como batería extremadamente baja o sobrecalentamiento de un motor), y el usuario no responde a las alertas en pantalla. La Jetson Orion toma el control de la situación y activa un protocolo de seguridad, deteniendo progresivamente la operación y mostrando en la interfaz que el sistema está en proceso de apagado seguro.   **Fallo en la Finalización de la Operación y Cierre del Sistema**   * La **Jetson Orion** envía el comando de cierre a la Raspberry Pi, pero no recibe confirmación de detención segura de los ESP32 de las patas. La interfaz de usuario muestra un mensaje indicando un error en el proceso de apagado. La Jetson Orion reintenta el apagado seguro hasta obtener respuesta o notifica al usuario de una posible falla en el hardware.   **Interfaz Congelada o Sin Respuesta**   * La **Jetson Orion** detecta que la interfaz de usuario ha dejado de responder debido a un fallo en el sistema. Realiza un reinicio automático de la interfaz sin interrumpir la operación de la plataforma y muestra un mensaje de disculpa al usuario cuando el sistema vuelve a la normalidad. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4. La plataforma debe integrar sistemas de control autónomo que le permitan tomar decisiones adaptativas en tiempo real basadas en la retroalimentación sensorial y las condiciones del entorno.** | | | |  |
| Imagen que contiene Logotipo  Descripción generada automáticamente | **Universidad Autónoma de Occidente**  **DESARROLLO DE UNA PLATAFORMA POLIVALENTE BIOINSPIRADA EN UN CUADRÚPEDO PARA INTERACCIÓN HUMANO-MAQUINA EN ENTORNOS COMPLEJOS** | | Rev.:  000 | |
| Title:  **CASO DE USO**  La plataforma debe integrar sistemas de control autónomo que le permitan tomar decisiones adaptativas en tiempo real basadas en la retroalimentación sensorial y las condiciones del entorno. | | Document:  CUR-001 | Page:  1 de 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Historic review | | | |
| Rev. | Change description | Author | Date |
| 001 | Document construction |  |  |
| 002 | Structural Change |  |  |

|  |  |
| --- | --- |
| **GENERAL INFORMATION** |  |
| **Actors** | Usuario, Jetson Orion, Raspberry Pi, Esp32 |
| **Propuse** | Permitir a la plataforma operar de manera autónoma, tomando decisiones en tiempo real para adaptarse a las condiciones del entorno. |
| **Summarize** | Este caso de uso describe cómo la plataforma utiliza su sistema de control autónomo para tomar decisiones adaptativas. |
| **Type** |  |

|  |  |
| --- | --- |
| **Curso normal de los eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| **Actor (Usuario)**: Activa el modo de operación autónoma a través de la interfaz de usuario en la Jetson Orion para que la plataforma tome decisiones basadas en retroalimentación sensorial. | La **Jetson Orion** activa todos los sistemas de percepción y comienza a recibir y analizar datos del entorno y el estado interno de la plataforma. |
| **Actor (Jetson Orion)**: Activa los sensores (lidar, cámara, IMU) y comienza a recibir información del entorno y del estado interno de la plataforma.  **Actor (ESP32 Sensorial)**: Envía datos en tiempo real a la Jetson Orion, incluyendo información de los encoders, IMU, y otros sensores de entorno. | La **Jetson Orion** utiliza los datos de sensores para evaluar obstáculos y el estado del terreno. Con la retroalimentación del ESP32 Sensorial, establece las condiciones iniciales de operación autónoma y ajusta la estrategia de navegación en función de la percepción del entorno. |
| **Actor (Jetson Orion):** Procesa los datos sensoriales y analiza las condiciones del entorno (obstáculos, terreno irregular, etc.). Genera un plan de acción o una nueva trayectoria según las condiciones detectadas y las necesidades de la operación. | La **Jetson Orion** calcula una trayectoria adaptativa y establece comandos específicos para abordar los obstáculos o terrenos irregulares identificados, enviándolos a la Raspberry Pi. |
| **Actor (Jetson Orion):** Envía instrucciones adaptativas a la Raspberry Pi para ajustar la velocidad, dirección o estabilidad en función de la retroalimentación recibida.  **Actor (Raspberry Pi):** Recibe los comandos de adaptación y distribuye los ajustes necesarios mediante comunicación CAN a cada ESP32 de las patas. | La **Raspberry Pi** distribuye los comandos recibidos a cada ESP32 de las patas para que ajusten los movimientos de los motores según la retroalimentación del entorno, garantizando una respuesta precisa. |
| **Actor (ESP32 de cada pata):** Calcula la cinemática inversa con los ajustes proporcionados y aplica los ángulos de motores necesarios para que las patas mantengan la estabilidad y respondan al entorno.  **Actor (Módulo de Potencia):** Regula la energía distribuida a los motores brushless, asegurando que el consumo sea eficiente según los comandos adaptativos. | Cada **ESP32 de las patas** realiza la cinemática inversa con los ajustes en tiempo real para mantener la estabilidad de la plataforma. El **Módulo de Potencia** distribuye la energía necesaria de manera eficiente, manteniendo la respuesta de los motores bajo control adaptativo. |
| **Actor (Jetson Orion):** Continúa monitoreando los datos del ESP32 Sensorial y ajusta la trayectoria o las decisiones operativas en tiempo real si detecta cambios en el entorno o condiciones no anticipadas. | La **Jetson Orion** analiza continuamente los datos sensoriales, ajustando el plan de acción en función de cambios en el entorno. Este monitoreo constante permite recalibraciones inmediatas, asegurando que la plataforma reaccione a condiciones inesperadas. |
| **Actor (Usuario):** Detiene el modo autónomo mediante la interfaz de usuario o la Jetson Orion determina que la operación ha concluido.  **Actor (Jetson Orion):** Envia un comando de cierre seguro a la Raspberry Pi para detener progresivamente el movimiento y retornar a un estado pasivo. | La **Jetson Orion** detiene progresivamente el modo autónomo, enviando un comando de cierre seguro a la Raspberry Pi, que a su vez apaga los motores de forma ordenada y pasa el sistema a un estado de espera seguro. |

|  |
| --- |
| **Curso alternativo de los eventos** |
| **Respuesta del sistema** |
| **Error en la Recepción de Datos Sensoriales**   * La **Jetson Orion** detecta un fallo en los datos de uno o varios sensores (por ejemplo, datos incompletos de IMU o lidar). Muestra una alerta en la interfaz de usuario e intenta restablecer la conexión con los sensores. Si el fallo persiste, ajusta el plan de navegación a un modo de baja velocidad y mayor precaución, dependiendo solo de los sensores activos.   **Detección de Obstáculo No Anticipado**   * La **Jetson Orion** identifica un obstáculo inesperado demasiado cercano para maniobrar en tiempo real. Detiene la locomoción de inmediato y recalcula una nueva trayectoria. Si no hay una ruta alternativa segura, notifica al usuario y permanece en modo de espera hasta recibir nuevas instrucciones.   **Desincronización en la Comunicación con ESP32 de las Patas**   * La **Raspberry Pi** detecta una desincronización en la comunicación CAN con uno o más ESP32 de las patas. Intenta reestablecer la sincronización mediante una reemisión de comandos adaptativos. Si no se logra sincronizar, la Jetson Orion muestra una advertencia en la interfaz y ajusta la locomoción para reducir la carga en las patas afectadas, permitiendo una operación limitada.   **Cambio Rápido en las Condiciones del Terreno**   * La **Jetson Orion** detecta una inclinación o irregularidad del terreno mayor a la esperada mediante datos de la IMU. Ajusta inmediatamente la posición y velocidad de cada pata para mantener la estabilidad, enviando comandos correctivos a la Raspberry Pi y reduciendo la velocidad de avance hasta estabilizar la plataforma.   **Fallo en el Cálculo de Cinemática Inversa**   * Uno de los **ESP32 de las patas** no puede calcular la cinemática inversa para los motores debido a una anomalía en los datos de entrada o sobrecarga de procesamiento. El ESP32 informa a la Raspberry Pi, que detiene temporalmente el movimiento de esa pata. La **Jetson Orion** ajusta la trayectoria para compensar y notifica al usuario sobre la limitación en la operación.   **Sobrecalentamiento en el Módulo de Potencia**   * El **Módulo de Potencia** detecta un sobrecalentamiento debido a alta demanda en los motores brushless. La **Jetson Orion** recibe la alerta y ajusta la operación para reducir la potencia distribuida a los motores, ralentizando el movimiento hasta que se estabilice la temperatura.   **Interrupción Crítica del Sistema Autónomo**   * La **Jetson Orion** detecta una condición crítica (por ejemplo, baja energía o error en la retroalimentación de los sensores) que impide continuar en modo autónomo de forma segura. Muestra una notificación en la interfaz y coloca la plataforma en un modo de pausa segura, aguardando nuevas instrucciones o la resolución del problema antes de continuar. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **5. Debe ser capaz de adaptarse a situaciones imprevistas o escenarios no programados, demostrando flexibilidad y toma de decisiones autónomas.** | | | |  |
| Imagen que contiene Logotipo  Descripción generada automáticamente | **Universidad Autónoma de Occidente**  **DESARROLLO DE UNA PLATAFORMA POLIVALENTE BIOINSPIRADA EN UN CUADRÚPEDO PARA INTERACCIÓN HUMANO-MAQUINA EN ENTORNOS COMPLEJOS** | | Rev.:  000 | |
| Title:  **CASO DE USO**  Debe ser capaz de adaptarse a situaciones imprevistas o escenarios no programados, demostrando flexibilidad y toma de decisiones autónomas. | | Document:  CUR-001 | Page:  1 de 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Historic review | | | |
| Rev. | Change description | Author | Date |
| 001 | Document construction |  |  |
| 002 | Structural Change |  |  |

|  |  |
| --- | --- |
| **GENERAL INFORMATION** |  |
| **Actors** | Usuario, Jetson Orion, Raspberry Pi, Esp32 |
| **Propuse** | Asegurar que la plataforma pueda manejar situaciones imprevistas de manera autónoma y eficiente. |
| **Summarize** | Este caso de uso describe cómo la plataforma maneja y se adapta a situaciones no programadas. |
| **Type** |  |

|  |  |
| --- | --- |
| **Curso normal de los eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| **Actor (Usuario)**: Activa el modo autónomo con capacidad de adaptación desde la interfaz en la Jetson Orion para que la plataforma responda de manera autónoma a situaciones no programadas. | La **Jetson Orion** inicializa todos los sistemas de percepción y análisis para monitorear el entorno y los datos del sistema en tiempo real, preparándose para tomar decisiones en escenarios no programados. |
| **Actor (Jetson Orion)**: Activa todos los sensores, incluyendo el lidar y la cámara, para recibir información continua del entorno. Analiza datos en tiempo real para detectar posibles obstáculos o cambios imprevistos.  **Actor (ESP32 Sensorial)**: Envía datos de encoders, IMU, y otros sensores de manera continua a la Jetson Orion, informando sobre el estado actual del terreno y de la plataforma. | La **Jetson Orion** procesa continuamente los datos sensoriales del ESP32, incluyendo información sobre obstáculos y condiciones del terreno, lo que permite detectar cambios en el entorno y reaccionar oportunamente. |
| **Actor (Jetson Orion)**: Identifica un cambio en el entorno (obstáculo inesperado, terreno resbaladizo, etc.) que no está programado en los patrones de navegación. Activa el módulo de toma de decisiones autónomas para analizar las alternativas. | La **Jetson Orion** identifica un cambio inesperado en el entorno y activa el sistema de toma de decisiones autónomas para evaluar la situación y determinar la mejor respuesta adaptativa. |
| **Actor (Jetson Orion):** Calcula una nueva trayectoria o estrategia de movimiento (por ejemplo, evitar el obstáculo o reducir velocidad) y envía estos comandos a la Raspberry Pi para que los ejecute en las patas. | La **Jetson Orion** genera un nuevo plan de acción que incluye ajustes en la velocidad, trayectoria o estabilidad, y envía estos comandos a la Raspberry Pi, permitiendo que la plataforma responda de manera adecuada a la situación imprevista. |
| **Actor (Raspberry Pi):** Recibe los comandos de adaptación y ajusta las configuraciones de cada ESP32 de las patas según la decisión de la Jetson Orion.  **Actor (ESP32 de cada pata):** Realiza la cinemática inversa para los motores con los ajustes requeridos y envía los ángulos de los motores brushless necesarios para la nueva trayectoria adaptada. | La **Raspberry Pi** distribuye los comandos adaptativos a cada ESP32 de las patas, garantizando que cada pata ejecute los movimientos necesarios para seguir la nueva trayectoria adaptada en tiempo real. |
| **Actor (ESP32 Sensorial):** Continúa monitoreando el estado de la plataforma y del entorno, enviando retroalimentación en tiempo real a la Jetson Orion.  **Actor (Jetson Orion):** Evalúa continuamente el éxito de la adaptación realizada y ajusta la estrategia si las condiciones del entorno cambian nuevamente. | Cada **ESP32 de las patas** ajusta los motores brushless según la nueva trayectoria, mientras el **ESP32 Sensorial** envía retroalimentación a la Jetson Orion para confirmar que la adaptación ha sido efectiva. |
| **Actor (Jetson Orion):** Mantiene el monitoreo del entorno y ajusta progresivamente el movimiento de la plataforma si detecta más cambios o situaciones imprevistas, manteniendo una toma de decisiones constante. | La **Jetson Orion** sigue evaluando las condiciones del entorno y ajusta los comandos adaptativos si se detectan nuevos cambios, demostrando flexibilidad y capacidad de adaptación continua. |
| **Actor (Usuario):** Finaliza el modo autónomo mediante la interfaz de usuario o la Jetson Orion determina que la operación ha concluido.  **Actor (Jetson Orion):** Envía una señal a la Raspberry Pi para cerrar progresivamente la operación de las patas y retomar un estado pasivo. | La **Jetson Orion** envía un comando de cierre seguro a la Raspberry Pi, que asegura el apagado progresivo de los motores de las patas y confirma la finalización de la operación. La interfaz de usuario muestra que el sistema ha vuelto a un estado pasivo. |

|  |
| --- |
| **Curso alternativo de los eventos** |
| **Respuesta del sistema** |
| **Fallo en la Recepción de Datos de Sensores Críticos**   * La **Jetson Orion** detecta una falta de datos de uno o varios sensores críticos (como lidar o IMU), lo que limita la percepción del entorno. Emite una alerta en la interfaz de usuario e intenta recuperar la conexión con los sensores. Si la falla persiste, ajusta el sistema para proceder en modo seguro, con velocidad reducida y precaución adicional.   **Cambio Extremo e Inesperado en el Terreno**   * La **Jetson Orion** identifica un cambio abrupto en el terreno (por ejemplo, inclinación pronunciada o superficie resbaladiza) que podría poner en riesgo la estabilidad. Detiene temporalmente el movimiento y recalcula una nueva estrategia de postura y ajuste de patas. Si el terreno sigue siendo inestable, permanece en espera y notifica al usuario, sugiriendo intervención manual.   **Obstáculo Demasiado Cercano para Maniobrar**   * La **Jetson Orion** detecta un obstáculo imprevisto muy cercano, lo que impide una maniobra de evasión en tiempo real. Activa un protocolo de parada inmediata para evitar colisiones y notifica al usuario sobre la presencia del obstáculo. Intenta recalcular la trayectoria y, si no es posible, espera instrucciones adicionales.   **Desincronización de los Comandos en los ESP32 de las Patas**   * La **Raspberry Pi** detecta una desincronización en la respuesta de uno o más ESP32 de las patas. Intenta reestablecer la sincronización enviando una señal de ajuste. Si no se logra resolver la desincronización, la Jetson Orion ajusta la locomoción para operar con las patas restantes, manteniendo un modo de adaptación limitada y notificando al usuario.   **Fallo en el Cálculo de la Nueva Trayectoria Adaptativa**   * La **Jetson Orion** no puede calcular una nueva trayectoria debido a limitaciones de procesamiento o datos insuficientes del entorno. En respuesta, mantiene la posición de la plataforma de manera segura y emite una alerta al usuario indicando la dificultad en la adaptación, sugiriendo posibles ajustes manuales.   **Sobrecalentamiento en Motores debido a Esfuerzo Continuo**   * La **Jetson Orion** recibe una alerta del Módulo de Potencia de que los motores brushless están operando cerca de su límite térmico debido a ajustes frecuentes. Reduce la velocidad de operación y limita las maniobras de alta carga. Notifica al usuario y adapta el sistema para asegurar la integridad de los motores mientras la temperatura desciende.   **Interrupción en el Modo Autónomo por Condiciones Críticas**   * La **Jetson Orion** detecta una condición crítica (como batería baja o error en el procesamiento sensorial) que imposibilita una operación segura en modo autónomo. Transfiere el sistema a un estado de espera segura y muestra una alerta en la interfaz de usuario, solicitando intervención manual antes de retomar la operación autónoma. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **6. Se requiere que la plataforma opere de manera eficiente desde el punto de vista energético para maximizar su autonomía y minimizar el impacto ambiental.** | | | |  |
| Imagen que contiene Logotipo  Descripción generada automáticamente | **Universidad Autónoma de Occidente**  **DESARROLLO DE UNA PLATAFORMA POLIVALENTE BIOINSPIRADA EN UN CUADRÚPEDO PARA INTERACCIÓN HUMANO-MAQUINA EN ENTORNOS COMPLEJOS** | | Rev.:  000 | |
| Title:  **CASO DE USO**  Se requiere que la plataforma opere de manera eficiente desde el punto de vista energético para maximizar su autonomía y minimizar el impacto ambiental. | | Document:  CUR-001 | Page:  1 de 1 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Historic review | | | |
| Rev. | Change description | Author | Date |
| 001 | Document construction |  |  |
| 002 | Structural Change |  |  |

|  |  |
| --- | --- |
| **GENERAL INFORMATION** |  |
| **Actors** | Usuario  Sistema de gestión de energía de la plataforma |
| **Propuse** | Asegurar que la plataforma funcione de manera eficiente y sostenible, optimizando el uso de energía. |
| **Summarize** | Este caso de uso describe cómo la plataforma gestiona su consumo energético para maximizar la autonomía y minimizar el impacto ambiental. |
| **Type** |  |

|  |  |
| --- | --- |
| **Curso normal de los eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| **Actor (Usuario):** Activa el modo de operación eficiente en la Jetson Orion a través de la interfaz de usuario para maximizar la autonomía de la batería. | La **Jetson Orion** activa modos de ahorro energético en todos los subsistemas y ajusta la configuración para operar con el mínimo consumo de energía necesario. |
| **Actor (Jetson Orion):** Monitorea los niveles de batería y calcula la carga de trabajo de cada componente. Define una estrategia de ahorro energético basada en el estado de la batería y los requisitos de operación.  **Actor (ESP32 Sensorial):** Envía datos del consumo energético de motores y otros sensores, permitiendo a la Jetson Orion ajustar las prioridades de energía en tiempo real. | La **Jetson Orion** analiza los datos de la batería y los reportes del ESP32 Sensorial, determinando la carga y ajustando prioridades de energía en tiempo real para cada componente. |
| **Actor (Jetson Orion):** Envía instrucciones de optimización a la Raspberry Pi, solicitando un ajuste en la frecuencia de operación de los motores y reducción de la velocidad cuando las condiciones lo permitan.  **Actor (Raspberry Pi):** Aplica los ajustes de energía solicitados por la Orion y distribuye comandos a los ESP32 de las patas. | La **Raspberry Pi** ajusta la frecuencia de operación de los motores y reduce la velocidad de locomoción según las instrucciones de ahorro energético de la Jetson Orion, manteniendo el equilibrio entre eficiencia y estabilidad. |
| **Actor (ESP32 de cada pata):** Ajusta la fuerza de los motores brushless, reduciendo el consumo de energía mientras mantiene la estabilidad en el movimiento.  **Actor (Módulo de Potencia):** Regula la energía proporcionada a los motores en base a los comandos del ESP32, limitando el pico de corriente cuando no es necesario. | Cada **ESP32 de las patas** reduce la potencia en los motores brushless de forma controlada, optimizando el consumo sin comprometer el movimiento.  El **Módulo de Potencia** adapta la corriente suministrada a los motores, evitando picos de consumo innecesarios y mejorando la eficiencia general del sistema. |
| **Actor (Jetson Orion):** Monitorea continuamente la carga restante de la batería y ajusta las estrategias de ahorro de energía de manera dinámica según el nivel de energía disponible. | La **Jetson Orion** actualiza el estado de la batería en tiempo real y ajusta la estrategia de operación conforme el nivel de carga disminuye, asegurando la mayor autonomía posible. |
| **Actor (Jetson Orion):** Envia una alerta al usuario cuando la batería alcanza un umbral crítico, sugiriendo reducir las funciones o detener la operación si es necesario. | La **Jetson Orion** envía alertas al usuario sobre el bajo nivel de energía y sugiere acciones para conservar la batería, como detener la operación o reducir funcionalidades no esenciales. |
| **Actor (Usuario):** Envía un comando para finalizar la operación o la Jetson Orion detecta que la batería está cerca de agotarse.  **Actor (Jetson Orion):** Ordena a la Raspberry Pi y al módulo de potencia que reduzcan progresivamente la energía y desconecten componentes no esenciales antes de apagar el sistema. | La **Jetson Orion** activa un proceso de apagado controlado, reduciendo gradualmente la energía a los motores y desconectando componentes secundarios. Envía una confirmación al usuario una vez que el sistema se encuentra en modo de bajo consumo o apagado seguro. |

|  |
| --- |
| **Curso alternativo de los eventos** |
| **Respuesta del sistema** |
| **Detección de Consumo Energético Anómalo**   * La Jetson Orion detecta un consumo excesivo en uno o varios componentes a través de los datos proporcionados por el ESP32 Sensorial y el Módulo de Potencia. Emite una alerta al usuario indicando que el sistema está consumiendo más energía de la esperada y reduce de inmediato la carga de trabajo de los componentes afectados para estabilizar el consumo.   **Fallos en la Comunicación con el Módulo de Potencia**   * La Jetson Orion pierde comunicación con el Módulo de Potencia o recibe lecturas inconsistentes sobre el suministro energético. Notifica al usuario y transfiere el sistema a un modo de operación mínima, apagando los subsistemas menos críticos hasta restablecer la comunicación.   **Disminución Rápida del Nivel de Batería**   * La Jetson Orion detecta una caída acelerada en el nivel de la batería. Ajusta la operación de la Raspberry Pi y los ESP32 de las patas para reducir el uso de energía de inmediato, limitando la velocidad y frecuencia de los motores para conservar batería. Envía una alerta al usuario y solicita una evaluación del estado de la batería si la disminución continúa.   **Sobrecalentamiento en Motores Brushless**   * Uno o más ESP32 de las patas detectan que los motores brushless están operando a temperaturas elevadas debido a alta demanda de potencia. El Módulo de Potencia reduce automáticamente el suministro a esos motores, y la Jetson Orion ajusta las maniobras de locomoción para disminuir la carga en las patas afectadas, protegiendo los componentes y reduciendo el consumo energético.   **Fallo en el Modo de Ahorro de Energía**   * La Jetson Orion no puede aplicar la configuración de ahorro energético correctamente debido a un fallo en la Raspberry Pi o en los ESP32 de las patas. En respuesta, la Orion prioriza el apagado de componentes secundarios y reduce las funciones no esenciales. Notifica al usuario sobre la limitación en el modo de ahorro de energía y mantiene el sistema en un estado de operación básica.   **Reducción Crítica del Nivel de Energía**   * La Jetson Orion detecta que la carga de la batería ha alcanzado un nivel crítico, donde el sistema no puede operar de forma segura. Inicia un apagado de emergencia, desactivando cada componente en secuencia para proteger la integridad de la batería y del sistema. Informa al usuario de la necesidad de recargar la batería y confirma una desconexión segura.   **Ajuste Fallido de Velocidad o Frecuencia de Componentes**   * La Raspberry Pi o los ESP32 de las patas no pueden ajustar correctamente la velocidad o frecuencia de los motores debido a un error en la configuración de ahorro energético. La Jetson Orion intenta reconfigurar los parámetros o transfiere el sistema a un modo de bajo consumo limitado, con una notificación al usuario sobre la necesidad de revisión. |