**PROYECTO FINAL DISEÑO Y CONSTRUCCIÓN DE PRODUCTOS DE SOFTWARE**

PLATAFORMA DE SERVICIO DE LIMPIEZA DE ROBOTS

**Por:**

Sebastián Olaya Pérez

Kevin Leonardo Arias Orrego

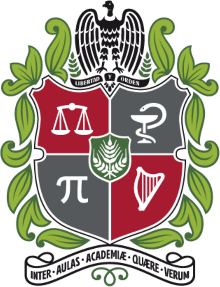
Juan Sebastián Zapata Echeverri

Profesor:

Albeiro Espinosa Bedoya

aespinos@unal.edu.co

28 de noviembre 2023

****

**Universidad Nacional de Colombia**

**Facultad de Minas**

**2023**

**Contenido**

[INTRODUCCIÓN 2](#_heading=h.gjdgxs)

[1. REQUISITOS 3](#_heading=h.30j0zll)

[2. DIAGRAMA DE CLASES (MODELO DEL DOMINIO) 4](#_heading=h.1fob9te)

[3. DIAGRAMA DE CASOS DE USO 5](#_heading=h.3znysh7)

[4. TABLAS DE ESPECIFICACIONES DE CASOS DE USO 5](#_heading=h.2et92p0)

[5. DIAGRAMAS DE ROBUSTEZ 15](#_heading=h.tyjcwt)

[6. DIAGRAMAS DE SECUENCIA 22](#_heading=h.3dy6vkm)

# INTRODUCCIÓN

El proyecto consiste en desarrollar una plataforma de servicio de robots en la nube que limpia habitaciones de hotel. El objetivo es que los robots puedan realizar la tarea de limpieza de forma eficiente y cooperativa, utilizando la plataforma como un medio de comunicación, coordinación y optimización. El proyecto se basa en el análisis y el modelado de un problema específico, utilizando técnicas y herramientas de ingeniería de software, como el modelo de dominio, el diagrama de casos de uso, las tablas de casos de uso, los diagramas de robustez, el diagrama de secuencias y el diagrama de clases. El proyecto pretende aplicar los principios y las prácticas de la programación orientada a objetos, y cumplir con los requisitos funcionales y no funcionales del sistema.

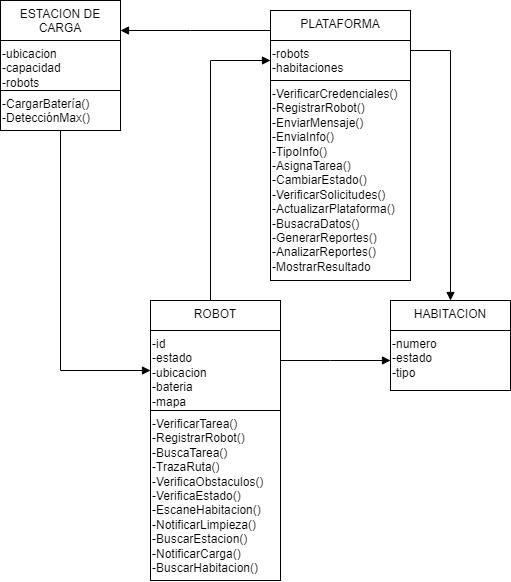
# 1. REQUISITOS

Para obtener los requisitos, se utilizó el contexto de la información que tiene el cliente, que describe el problema y el objetivo del sistema. A partir de esta información, se extrajeron los requisitos funcionales, que definen las funcionalidades y las restricciones que debe tener el programa, y los requisitos no funcionales, que definen las propiedades y características del sistema que no están directamente relacionadas con su funcionalidad, sino con su calidad y desempeño.

* El programa debe permitir el registro e identificación de los robots que se conectan a la plataforma. Este requisito se relaciona con el caso de uso Registrar un robot y la clase Robot, que tiene los atributos id, nombre, tipo y estado.
* El programa debe asignar una tarea de limpieza a cada robot según la disponibilidad y la prioridad de las habitaciones del hotel. Este requisito se relaciona con el caso de uso Asignar una tarea a un robot y las clases Plataforma, Robot y Habitación, que tienen las asociaciones asignar, realizar y limpiar, respectivamente.
* El programa debe proporcionar a los robots la ubicación de las habitaciones y las estaciones de carga que les corresponden. Este requisito se relaciona con los casos de uso Recibir una asignación de la plataforma, Navegar a una habitación asignada y Navegar a una estación de carga, y las clases Robot, Habitación y Estación de carga, que tienen los atributos ubicación y distancia.
* El programa debe monitorear el desempeño de los robots y el estado de las habitaciones en tiempo real. Este requisito se relaciona con el caso de uso Monitorear el desempeño de los robots y las clases Plataforma, Robot y Habitación, que tienen los atributos desempeño, nivel de batería, tiempo de limpieza y estado de limpieza.
* El programa debe facilitar la comunicación y la cooperación entre los robots que comparten la misma tarea, permitiéndoles intercambiar información relevante. Este requisito se relaciona con los casos de uso Descubrir otros robots con la misma tarea y Compartir información con robots, y la clase Robot, que tiene los atributos tarea, compañeros y mensajes.
* El programa debe optimizar el proceso de limpieza con múltiples robots, evitando duplicidades y conflictos. Este requisito se relaciona con el caso de uso Cambiar estado de habitación y las clases Plataforma, Robot y Habitación, que tienen las restricciones de unicidad, exclusividad y consistencia.
* El programa debe cumplir con los requisitos no funcionales de seguridad, rendimiento, escalabilidad y portabilidad. Estos requisitos se refieren a las propiedades y características del sistema que no están directamente relacionadas con su funcionalidad, sino con su calidad y desempeño.

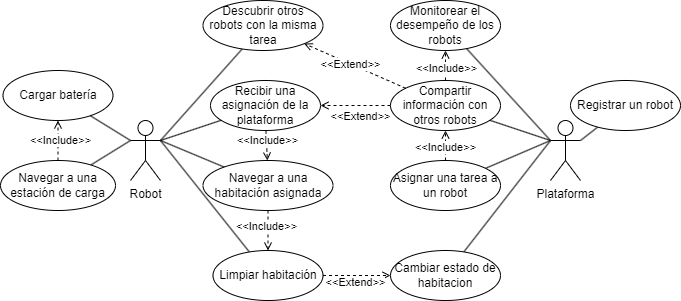
# 2. DIAGRAMA DE CLASES (MODELO DEL DOMINIO)

Para crear el diagrama de clases, se utilizó el modelo de dominio en fase temprana, que identificó las entidades, atributos, relaciones y restricciones del problema. Luego, se completó el modelo de dominio después de haber pasado por la creación de los casos de uso, que especificaron los escenarios y las interacciones entre los actores y el sistema. El diagrama de clases reflejó la estructura y el comportamiento del dominio del problema, y sirvió como base para el diseño del software orientado a objetos.



# 3. DIAGRAMA DE CASOS DE USO

Para crear el diagrama de casos de uso, se utilizó el modelo de dominio en fase temprana, que proporcionó los conceptos y el vocabulario clave del problema. También se utilizaron los requisitos pasados anteriormente, que definieron las funcionalidades y las restricciones que debía tener el programa. El diagrama de casos de uso mostró los actores, los casos de uso y las relaciones entre ellos, y describió los objetivos y las expectativas de los usuarios del sistema.



# 4. TABLAS DE ESPECIFICACIONES DE CASOS DE USO

Para crear las tablas de casos de uso, se utilizó el diagrama de casos de uso, que identificó los casos de uso y sus relaciones. Cada tabla de caso de uso detalló la descripción, las precondiciones, las postcondiciones, el flujo normal y el flujo alternativo de un caso de uso, y explicó los pasos y las acciones que se realizaban en cada escenario, tanto cuando todo salía bien como cuando ocurría algún problema.

| **Nombre** | Registrar un robot | **ID** | 001 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a la plataforma registrar un nuevo robot. | | |
| **Actor** | Plataforma | | |
| **Precondición** | El robot tiene una identificación y una clave de acceso válidas. | | |
| **Postcondición** | El robot queda registrado en la plataforma y puede acceder a sus servicios. | | |
| **Flujo Normal** | 1. El robot envía una solicitud de registro a la plataforma, incluyendo su identificación y su clave de acceso.  2. La plataforma recibe la solicitud y verifica la identificación y la clave de acceso del robot.  3. La plataforma confirma el registro y envía una respuesta al robot, indicando que puede acceder a sus servicios.  4. El robot recibe la respuesta y queda registrado en la plataforma. | | |
| **Secuencia alternativa** | - Si la identificación o la clave de acceso del robot son inválidas, la plataforma rechaza el registro y envía una respuesta al robot, indicando el motivo del rechazo. El robot recibe la respuesta y puede intentar registrarse de nuevo con datos válidos.  - Si la plataforma no está disponible o hay algún problema de comunicación, el robot no recibe la respuesta y puede intentar registrarse de nuevo más tarde. | | |

| **Nombre** | Descubrir otros robots con la misma tarea | **ID** | 002 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a un robot encontrar a otros robots que también están limpiando las habitaciones del hotel. | | |
| **Actor** | Robot | | |
| **Precondición** | El robot está registrado en la plataforma y tiene asignada una tarea de limpieza de habitaciones. | | |
| **Postcondición** | El robot conoce a otros robots que comparten la misma tarea y puede cooperar con ellos. | | |
| **Flujo Normal** | 1. El robot envía una solicitud de obtención de información a la plataforma, indicando su identificación y su tarea.  2. La plataforma recibe la solicitud y busca en su lista de robots los que tienen la misma tarea que el solicitante.  3. La plataforma envía una respuesta al robot, incluyendo la identificación, la ubicación y el estado de los robots que comparten la misma tarea.  4. El robot recibe la respuesta y puede comunicarse con los otros robots directamente o a través de la plataforma. | | |
| **Secuencia alternativa** | - Si el robot no tiene asignada una tarea de limpieza de habitaciones, la plataforma rechaza la solicitud y envía una respuesta al robot, indicando el motivo del rechazo. El robot recibe la respuesta y puede solicitar una asignación de la plataforma.  - Si no hay otros robots que compartan la misma tarea que el solicitante, la plataforma envía una respuesta al robot, indicando que no hay otros robots disponibles. El robot recibe la respuesta y puede continuar con su tarea de forma individual.  - Si la plataforma no está disponible o hay algún problema de comunicación, el robot no recibe la respuesta y puede intentar descubrir a otros robots más tarde. | | |

| **Nombre** | Compartir información con robots | **ID** | 003 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a la plataforma enviar o recibir información relevante para la tarea de limpieza de habitaciones, como el estado de las habitaciones, la ubicación de los robots, el nivel de batería, el tiempo de limpieza, etc. | | |
| **Actor** | Plataforma | | |
| **Precondición** | La plataforma tiene registrados a robots que comparten la misma tarea de limpieza de habitaciones. | | |
| **Postcondición** | La plataforma y los robots tienen información actualizada sobre la tarea y pueden optimizar su desempeño. | | |
| **Flujo Normal** | 1. La plataforma recibe una solicitud de información de un robot, indicando el tipo de información que necesita.  2. La plataforma busca en su lista de robots o de habitaciones la información solicitada.  3. La plataforma envía una respuesta con la información solicitada al robot.  4. El robot recibe la respuesta y la utiliza para su propósito. | | |
| **Secuencia alternativa** | - Si la plataforma detecta un cambio en la información que afecta a la tarea de limpieza de habitaciones, como el estado de una habitación, la ubicación de un robot, el nivel de batería, etc., la plataforma envía una notificación con la información actualizada a los robots que se vean afectados por el cambio. Los robots o el comandante reciben la notificación y la utilizan para ajustar su comportamiento. | | |

| **Nombre** | Recibir una asignación de la plataforma | **ID** | 004 |
| --- | --- | --- | --- |
| **Descripción** | El robot recibe una tarea de limpiar una habitación específica de la plataforma. | | |
| **Actor** | Robot | | |
| **Precondición** | El robot está registrado en la plataforma y tiene una batería suficiente para realizar la tarea. | | |
| **Postcondición** | El robot tiene una habitación asignada para limpiar. | | |
| **Flujo Normal** | 1. El robot solicita a la plataforma una tarea disponible.  2. La plataforma envía al robot una tarea de limpiar una habitación específica.  3. El robot acepta la tarea y confirma a la plataforma. | | |
| **Secuencia alternativa** | - Si la plataforma no tiene ninguna tarea disponible para el robot, la plataforma envía una respuesta al robot, indicando que no hay ninguna tarea asignada. El robot recibe la respuesta y puede solicitar otra asignación más tarde o cargar su batería, utilizando el caso de uso [Cargar batería].  - Si el robot no puede limpiar la habitación asignada por algún motivo, como que la habitación está ocupada, que la puerta está cerrada, que hay algún desperfecto, etc., el robot informa a la plataforma del problema y solicita una nueva asignación. La plataforma recibe el informe y cambia el estado de la habitación a pendiente de revisión, utilizando el caso de uso [Cambiar estado de habitación]. La plataforma asigna una nueva tarea al robot. | | |

| **Nombre** | Navegar a una habitación asignada | **ID** | 005 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a un robot desplazarse desde su ubicación actual hasta una habitación que le ha sido asignada por la plataforma. | | |
| **Actor** | Robot | | |
| **Precondición** | El robot ha recibido una asignación de la plataforma y tiene una batería suficiente para realizar la tarea. | | |
| **Postcondición** | El robot llega a la habitación asignada | | |
| **Flujo Normal** | 1. El robot consulta su atributo mapa para encontrar la ruta más corta y segura hasta la habitación asignada, teniendo en cuenta el número del piso y el tipo de habitación.  2. El robot se desplaza por el hotel siguiendo la ruta indicada por el mapa, utilizando sus sensores y actuadores para evitar obstáculos y colisiones con otros robots o personas.  3. El robot llega a la habitación asignada y verifica su número y su estado.  4. Si la habitación está desocupada, el robot entra y se prepara para limpiarla. | | |
| **Secuencia alternativa** | - Si el robot no tiene el atributo mapa o el mapa está incompleto o desactualizado, el robot solicita a la plataforma que le envíe el mapa o la información necesaria para llegar a la habitación asignada. La plataforma recibe la solicitud y envía la respuesta al robot. El robot recibe la respuesta y actualiza su atributo mapa o sigue las instrucciones recibidas.  - Si el robot encuentra un obstáculo o una colisión en su ruta a la habitación asignada, el robot evita el obstáculo o la colisión y busca una ruta alternativa, utilizando su atributo mapa. El robot informa a la plataforma del incidente y continúa con su tarea.  - Si la habitación está ocupada, el robot solicita otra habitación disponible a la plataforma. | | |

| **Nombre** | Limpiar habitación | **ID** | 006 |
| --- | --- | --- | --- |
| **Descripción** | Este es un caso de uso que permite a un robot realizar la tarea de limpiar una habitación asignada por la plataforma. | | |
| **Actor** | Robot | | |
| **Precondición** | El robot ha recibido una asignación de la plataforma y ha llegado a la habitación asignada. | | |
| **Postcondición** | El robot ha limpiado la habitación y ha cambiado su estado a limpio. | | |
| **Flujo Normal** | 1. El robot verifica el estado de la habitación, comprobando si está ocupada, cerrada o sucia.  2. El robot escanea la habitación y detecta las áreas que requieren limpieza.  3. El robot selecciona el equipo de limpieza adecuado según el tipo de superficie y suciedad.  4. El robot informa a la plataforma y cambia el estado de la habitación a limpia.  5. El robot cambia su estado a disponible y espera una nueva asignación de la plataforma. | | |
| **Secuencia alternativa** | - Si la habitación está ocupada o cerrada, el robot informa a la plataforma del problema y solicita una nueva asignación.  - Si el robot no tiene el equipo de limpieza adecuado, informa a la plataforma y solicita el equipo necesario.  - Si el robot encuentra algún desperfecto o anomalía en la habitación, como una mancha difícil de quitar, un objeto roto, un olor extraño, etc., el robot informa a la plataforma del problema y solicita una nueva asignación. La plataforma recibe el informe y cambia el estado de la habitación a pendiente de revisión. | | |

| **Nombre** | Cargar batería | **ID** | 007 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a un robot recargar su atributo batería cuando está bajo un nivel mínimo. | | |
| **Actor** | Robot | | |
| **Precondición** | El robot está en una estación de carga disponible y tiene un nivel de batería bajo. | | |
| **Postcondición** | El robot tiene una batería por encima de un nivel óptimo y está listo para recibir una tarea o completarla. | | |
| **Flujo Normal** | 1. El robot se conecta a la estación de carga mediante un cable o un sistema inalámbrico.  2. El robot inicia el proceso de carga de su batería.  3. El robot monitorea el nivel de su batería durante la carga.  4. El robot detecta que su nivel de batería ha alcanzado un nivel óptimo.  5. El robot finaliza el proceso de carga de su batería y se desconecta de la estación de carga.  6. El robot informa a la plataforma que ha terminado la carga y que la estación de carga está disponible. | | |
| **Secuencia alternativa** | - Si no hay ninguna estación de carga disponible o el robot no puede acceder a ella por algún motivo, el robot informa a la plataforma del problema y espera una solución o una nueva asignación. | | |

| **Nombre** | Asignar una tarea a un robot | **ID** | 008 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a la plataforma enviar una tarea específica a un robot, cómo limpiar una habitación determinada. | | |
| **Actor** | Plataforma | | |
| **Precondición** | La plataforma tiene registrados a los robots que comparten la misma tarea de limpieza de habitaciones y tiene una lista de habitaciones disponibles para asignar. | | |
| **Postcondición** | La plataforma asigna una tarea a un robot y recibe su confirmación. | | |
| **Flujo Normal** | 1. La plataforma recibe una solicitud de asignación de un robot, indicando su identificación y su ubicación.  2. La plataforma asigna la tarea al robot, teniendo en cuenta la eficiencia, la equidad y la cooperación entre los robots.  3. La plataforma envía una respuesta al robot, indicando el número de la habitación que debe limpiar y la ruta más corta y segura para llegar a ella.  4. El robot recibe la respuesta y confirma la asignación, indicando que va a ejecutar la tarea.  5. La plataforma recibe la confirmación del robot y actualiza su lista de robots y de habitaciones disponibles. | | |
| **Secuencia alternativa** | - Si la plataforma no recibe una solicitud de asignación de un robot, la plataforma puede asignar una tarea a un robot de forma proactiva, teniendo en cuenta la eficiencia, la equidad y la cooperación entre los robots.  - Si la plataforma no tiene ninguna tarea disponible para asignar, la plataforma envía una respuesta al robot, indicando que no hay ninguna tarea asignada. El robot recibe la respuesta y puede solicitar otra asignación más tarde o cargar su batería, utilizando el caso de uso [Cargar batería].  - Si el robot no confirma la asignación o rechaza la tarea por algún motivo, la plataforma puede asignar la misma tarea a otro robot o buscar una tarea alternativa para el robot que rechazó la asignación. | | |

| **Nombre** | Monitorear el desempeño de los robots | **ID** | 009 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a la plataforma observar el comportamiento de los robots y evaluar su rendimiento, como el tiempo de limpieza, el número de habitaciones limpiadas, el consumo de energía. | | |
| **Actor** | Plataforma | | |
| **Precondición** | La plataforma tiene registrados a los robots que comparten la misma tarea de limpieza de habitaciones y tiene una lista de habitaciones disponibles para asignar. | | |
| **Postcondición** | La plataforma tiene información actualizada sobre el desempeño de los robots y puede mejorar su servicio. | | |
| **Flujo Normal** | 1. La plataforma recibe información de los robots o la solicita periódicamente, utilizando el caso de uso [Compartir información con robots].  2. La plataforma almacena la información en su atributo reportes, que es una colección de datos sobre el desempeño de los robots.  3. La plataforma analiza la información y calcula indicadores de rendimiento, como el tiempo promedio de limpieza, el número total de habitaciones limpiadas, el consumo promedio de energía, etc.  4. La plataforma genera reportes o alertas sobre el desempeño de los robots. | | |
| **Secuencia alternativa** | - Si la plataforma detecta un problema o una anomalía en el desempeño de los robots, como un tiempo de limpieza excesivo, un número de habitaciones limpiadas insuficiente, un consumo de energía elevado, etc., la plataforma envía una alerta al robot o al responsable del problema, indicando el motivo de la alerta y la posible solución. | | |

| **Nombre** | Cambiar estado de habitación | **ID** | 010 |
| --- | --- | --- | --- |
| **Descripción** | Este caso de uso permite a la plataforma modificar el atributo estado de una habitación según las acciones que se realicen sobre ella. | | |
| **Actor** | Plataforma | | |
| **Precondición** | La plataforma tiene una lista de habitaciones disponibles para asignar. | | |
| **Postcondición** | La plataforma cambia el estado de una habitación y lo informa a los robots que se vean afectados por el cambio. | | |
| **Flujo Normal** | 1. La plataforma recibe una acción que implica un cambio de estado de una habitación, una solicitud de reserva de un cliente o una detección de un problema.  2. La plataforma busca en su lista de habitaciones la habitación que corresponde a la acción recibida.  3. La plataforma cambia el atributo estado de la habitación según la acción recibida, como limpia, ocupada o pendiente de revisión.  4. La plataforma envía una notificación con el cambio de estado de la habitación a los robots o al comandante que se vean afectados por el cambio, utilizando el caso de uso [Compartir información con robots]. | | |
| **Secuencia alternativa** | - Si la plataforma no encuentra la habitación que corresponde a la acción recibida, la plataforma envía una alerta indicando que hay un error en la información recibida y solicita una corrección o una confirmación. | | |

| **Nombre** | Navegar a una estación de carga | **ID** | 011 |
| --- | --- | --- | --- |
| **Descripción** | El robot se dirige a una estación de carga disponible cuando detecta que su nivel de batería es bajo. | | |
| **Actor** | Robot | | |
| **Precondición** | El robot tiene una batería por debajo de un nivel mínimo y no tiene ninguna tarea asignada. | | |
| **Postcondición** | El robot llega a una estación de carga disponible. | | |
| **Flujo Normal** | 1. El robot detecta que su nivel de batería es bajo.  2. El robot solicita a la plataforma la ubicación de una estación de carga disponible.  3. La plataforma envía al robot la ubicación de una estación de carga disponible.  4. El robot se dirige a la estación de carga siguiendo la ruta más corta y evitando obstáculos.  5. El robot llega a la estación de carga y se detiene. | | |
| **Secuencia alternativa** | - Si el robot no recibe la ubicación de una estación de carga disponible, se repite el paso 2.  - Si el robot encuentra un obstáculo en su ruta, busca una ruta alternativa y se repite el paso 4.  - Si el robot no puede llegar a la estación de carga por algún motivo, informa a la plataforma y solicita otra estación de carga disponible. | | |

# 5. DIAGRAMAS DE ROBUSTEZ

Para crear los diagramas de robustez, se utilizó cada flujo de cada tabla de caso de uso, incluyendo los flujos alternativos. Cada diagrama de robustez representó gráficamente los elementos, las acciones y las interacciones que intervienen en un flujo de un caso de uso, y clasificó los elementos en tres tipos: actores, entidades y controladores. Los diagramas de robustez ayudaron a validar y verificar los casos de uso, y a identificar las responsabilidades y los atributos de las clases del sistema.

## 1. Registrar un robot ID 001

## 2. Descubrir otros robots con la misma tarea ID 002

## 3. Compartir información con robots ID 003

## 4. Recibir una asignación de la plataforma ID 004

## 5. Navegar a una habitación asignada ID 005

## 6. Limpiar habitación ID 006

## 7. Cargar batería ID 007

## 8. Asignar una tarea a un robot ID 008

## 9. Monitorear el desempeño de los robots ID 009

## 10. Cambiar estado de habitación ID 010

## 11. Navegar a una estación de carga ID 011

# 6. DIAGRAMAS DE SECUENCIA

Para crear el diagrama de secuencias, se utilizó el diagrama de robustez, que proporcionó los elementos, las acciones y las interacciones de un flujo de un caso de uso. El diagrama de secuencias mostró la secuencia temporal de los mensajes que se intercambiaban entre los elementos, y especificó los parámetros, los retornos y las condiciones de los mensajes. El diagrama de secuencias completó el planteamiento y el modelado de un software, y facilitó la implementación y el código del sistema.

## 1. Registrar un robot ID 001

## 2. Descubrir otros robots con la misma tarea ID 002

## 3. Compartir información con robots ID 003

## 4. Recibir una asignación de la plataforma ID 004

## 5. Navegar a una habitación asignada ID 005

## 6. Limpiar habitación ID 006

## 7. Cargar batería ID 007

## 8. Asignar una tarea a un robot ID 008

## 9. Monitorear el desempeño de los robots ID 009

## 10. Cambiar estado de habitación ID 010

## 11. Navegar a una estación de carga ID 011