

INSTITUTO TECNOLÓGICO DE COSTA RICA

PROYECTO DE DISEÑO DE INGENIERÍA EN COMPUTADORES  
CE-5302

---

## **Documento de Diseño - KROTIC Architecture Upgrade**

---

**Estudiante**

Marcelo Sánchez Solano

**Carné**

2016115728

# Índice

<b>1. Información general del documento</b>	<b>2</b>
<b>2. Historial de cambios</b>	<b>2</b>
<b>3. Introducción</b>	<b>2</b>
3.1. Propósito . . . . .	2
3.2. Alcance . . . . .	2
3.3. Contexto . . . . .	3
3.4. Resumen . . . . .	3
<b>4. Glosario</b>	<b>3</b>
<b>5. Interesados</b>	<b>4</b>
<b>6. Perspectivas de diseño</b>	<b>4</b>
6.1. Contexto . . . . .	4
6.2. Composición . . . . .	5
6.3. Lógica . . . . .	7
6.4. Dependencias . . . . .	7
6.5. Información . . . . .	8
6.6. Uso de patrones . . . . .	8
6.7. Interfaces . . . . .	9
6.8. Interfaces de usuario . . . . .	10
6.9. Estructura . . . . .	10
6.10. Interacción . . . . .	10
6.11. Dinámica de estados . . . . .	11
6.12. Algoritmos . . . . .	11
6.13. Recursos . . . . .	11
<b>7. Apéndices</b>	<b>12</b>
7.1. Apéndice A. Alternativas de diseño . . . . .	12
7.1.1. Propuesta de dos niveles simple . . . . .	12
7.1.2. Propuesta de dos niveles con nivel inferior separado . . . . .	13
7.1.3. Propuesta de dos niveles con unidad intermediaria de control . . . . .	13
7.2. Apéndice B. Comparación de propuestas de diseño . . . . .	15
7.2.1. Selección de propuesta . . . . .	15
7.3. Apéndice C. Tabla de tarjetas de desarrollo candidatas para implementación	16

# 1. Información general del documento

Cuadro 1: Información general del documento.

<b>Organización</b>	Laboratorio LuTec.
<b>Fecha de la versión</b>	16/03/2021
<b>Estatus</b>	Preliminar.
<b>Autor(es)</b>	Marcelo Sánchez Solano.

# 2. Historial de cambios

Cuadro 2: Tabla de cambios del documento de diseño.

<b>Versión</b>	<b>Fecha de versión</b>	<b>Descripción de cambios</b>
1.0.0	09/03/2021	Creación inicial de estructura del documento y redacción de introducción.
1.0.1	10/03/2021	Confección de diagramas de contexto y composición.
1.0.2	11/03/2021	Confección de diagramas de dependencias, patrones, secuencia y estructural.
1.0.3	13/03/2021	Diagramas de propuestas de solución.
1.0.4	16/03/2021	Actualización de diagramas de solución y comparación de propuestas.

# 3. Introducción

## 3.1. Propósito

El sistema KROTIC tiene como objetivo controlar todas las funciones que realiza un robot, para que niños de zonas alejadas puedan incursionar en el mundo de la robótica incluso si no están de forma presencial con el robot. Dado el contexto de la pandemia mundial debido al COVID-19, el sistema se puede utilizar remotamente de modo que no se interrumpa el aprendizaje de conceptos de robótica.

## 3.2. Alcance

El funcionamiento de KROTIC se divide en dos partes: la parte de control de interacción y transmisión de datos, que es la encargada de comunicarse con el niño remotamente. La otra parte se encarga de la ejecución de las funciones que son programadas por los estudiantes.

### 3.3. Contexto

Para la realización de este proyecto, el sistema KROTIC debe actualizarse debido al surgimiento de nuevas necesidades que el diseño anterior no soporta completamente. Una de las mayores funcionalidades que se derivan es la necesidad de interacción remota, dada la emergencia sanitaria mundial. Por eso se requiere que las funcionalidades sean divididas de modo que la experiencia de aprendizaje no se vea afectada.

### 3.4. Resumen

El robot KROTIC es un robot de propósito educativo, dirigido a niños de zonas alejadas donde los recursos tecnológicos son limitados. El robot sirve como base para que los estudiantes de estas zonas puedan incursionar desde edades tempranas en la robótica y puedan ejercitar los conocimientos de manera práctica.

Los niños aunque no estén utilizando el robot físicamente, pueden ser capaces de desarrollar sus programas y ejecutarlos remotamente. Además de poder ver el resultado de su programa a través de transmisión desde el robot hacia el estudiante, con el objetivo de que los niños tengan retroalimentación sobre las soluciones que fueron programadas.

## 4. Glosario

1. **Fábrica abstracta.** Patrón de diseño creacional que produce familias de objetos relacionados (1).
2. **Patrón de diseño.** Soluciones generales a problemas comunes dentro del contexto de *software* (2).
3. **Retroalimentación.** La salida o resultado de una parte del sistema es reintroducido de vuelta.

## Referencias

- [1] Abstract Factory. Tomado de: <https://refactoring.guru/es/design-patterns/abstract-factory>
- [2] Patrones de diseño. Tomado de: <https://refactoring.guru/es/design-patterns>

## 5. Interesados

En la tabla 3 se muestran los actores interesados en el proyecto junto con los beneficios que se obtienen por la realización del proyecto y los factores de interés.

Cuadro 3: Valor e interés de cada interesado del proyecto.

Interesado	Valor	Interés
Estudiante	Observar el programa desarrollado por ellos mismos en funcionamiento de forma remota.	Aprender sobre principios básicos de robótica y programación.
Profesores	Permitir enseñar conceptos de robótica y programación de forma aplicada.	Evaluar los temas estudiados a través de ejercicios prácticos.
Asistente de laboratorio	Controlar el sistema para el funcionamiento de KROTIC	Asistir en el proceso de aprendizaje de los niños desde el laboratorio.
Desarrollador de la arquitectura	Provee de las herramientas necesarias para que los estudiantes puedan desarrollar y ejecutar sus programas de forma remota.	Aprender sobre los componentes de un sistema de robótica y circuitos empotrados.
Instituto Tecnológico de Costa Rica	Contribuir con el aprendizaje de principios de robótica en zonas lejanas y recursos limitados.	Promover el aprendizaje de temas relevantes actuales, como lo es la robótica y la programación.

## 6. Perspectivas de diseño

### 6.1. Contexto

El diagrama de la figura 1 muestra todas las interacciones del diseño de la arquitectura del robot KROTIC. Por un lado se tiene el aspecto de personal, que consta del asistente de laboratorio; que será el encargado de realizar la configuración inicial del robot. El asistente es el único usuario que tiene interacción directa con la arquitectura del robot.

Por otro lado, se tienen los aspectos de operación y funcionalidad. A continuación se explican las partes involucradas:

1. **Sensores.** Cuatro tipos de sensores que pueden ser utilizados por separado y a conveniencia por los estudiantes que crean el programa que se ejecuta en la arquitectura.
2. **Luces.** Similar a los sensores, las luces responden a acciones estipuladas en el programa de los estudiantes. Los programas soportan las acciones básicas de encendido y apagado.

3. **Módulo de conexión inalámbrica.** Este componente es el encargado de la transmisión y recepción de datos de los estudiantes. Se utiliza para dar la retroalimentación a los estudiantes sobre el resultado del programa enviado.
4. **Cámara.** Componente propio del robot, que está destinado a capturar la vista en primera persona de las acciones que realiza el robot para luego ser transmitida a los estudiantes.
5. **Servos.** Son los componentes que controlan las acciones de movimiento del robot.

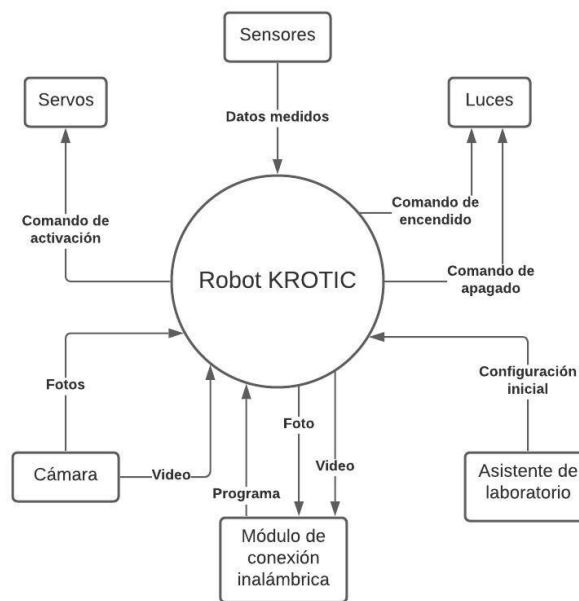


Figura 1: Diagrama de contexto del sistema.

## 6.2. Composición

En este caso, se observa que en la figura 2 se desglosan las dos partes de la arquitectura que manejan el uso de datos, comunicación y funcionalidad. A continuación se explican las partes del diagrama, sin incluir aquellas partes que ya fueron explicadas en el diagrama de contexto.

- **Unidades de procesamiento.** Son las encargadas de manejar comunicación interna, es decir, transmisión de datos entre las diferentes unidades y se encargan de manejar los dispositivos conectados a ellas.
- **Módulo de protección.** Protección contra eventos que pongan en peligro la integridad y funcionalidad del robot.
- **Sensores.** Captura de diferentes factores del entorno.

- **Nivel superior.** Controla la cámara, transmisión de fotos y videos y recepción de datos del estudiante.
- **Nivel inferior.** Encargado de controlar sensores, componentes de movimiento y protección de los dispositivos.

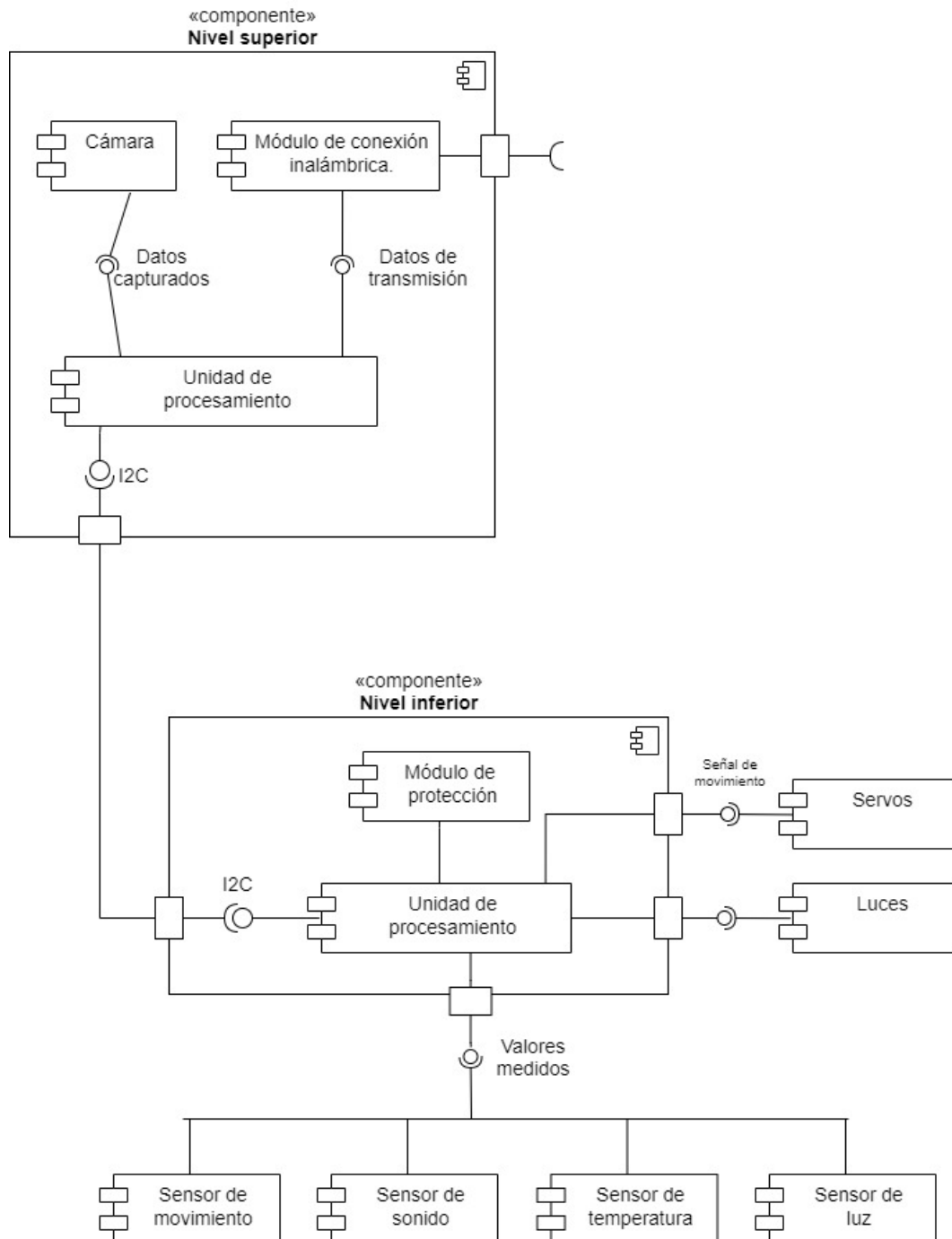


Figura 2: Diagrama de composición del sistema.

### 6.3. Lógica

No aplica.

### 6.4. Dependencias

En el caso del modelo de dependencias de la figura 3, aparte de los componentes ya visualizados en el diagrama de composición (figura 2), se especifican módulos de manejo de funciones internas de las placas de desarrollo que se utilicen:

1. **Receptor de programa.** Unidad que se basa la comunicación inalámbrica para recibir los programas de los estudiantes.
2. **Unidad de planificación.** Al recibir el programa, procesará las instrucciones del programa y dividirá las tareas. Por ejemplo, las instrucciones se podrían dividir en instrucciones más pequeñas y atómicas que resultan en un manejo más eficiente de los recursos computacionales. Una vez que se planifiquen, se comunican las acciones finales al nivel de funcionalidad.
3. **Unidad de procesamiento multimedia.** Módulo de procesamiento de los datos tomados por la cámara.
4. **Módulo de comunicación interna.** La comunicación interna hace referencia a la conexión entre el nivel superior con el inferior. Este módulo comunica las acciones que se deben realizar con respecto a movimiento, mediciones y control de luces.
5. **Módulo de protección.** Consiste en una unidad de monitoreo de variables que puedan poner en peligro a los componentes. Se incluyen situaciones como baja batería y sobrecalentamiento del dispositivo principal.
6. **Módulo de temporización.** Encargado de temporizar acciones y es activado a través de instrucciones de los programas de estudiantes.
7. **Módulo de ejecución.** Encargado de controlar todos los componentes funcionales externos, como sensores, servos y luces.



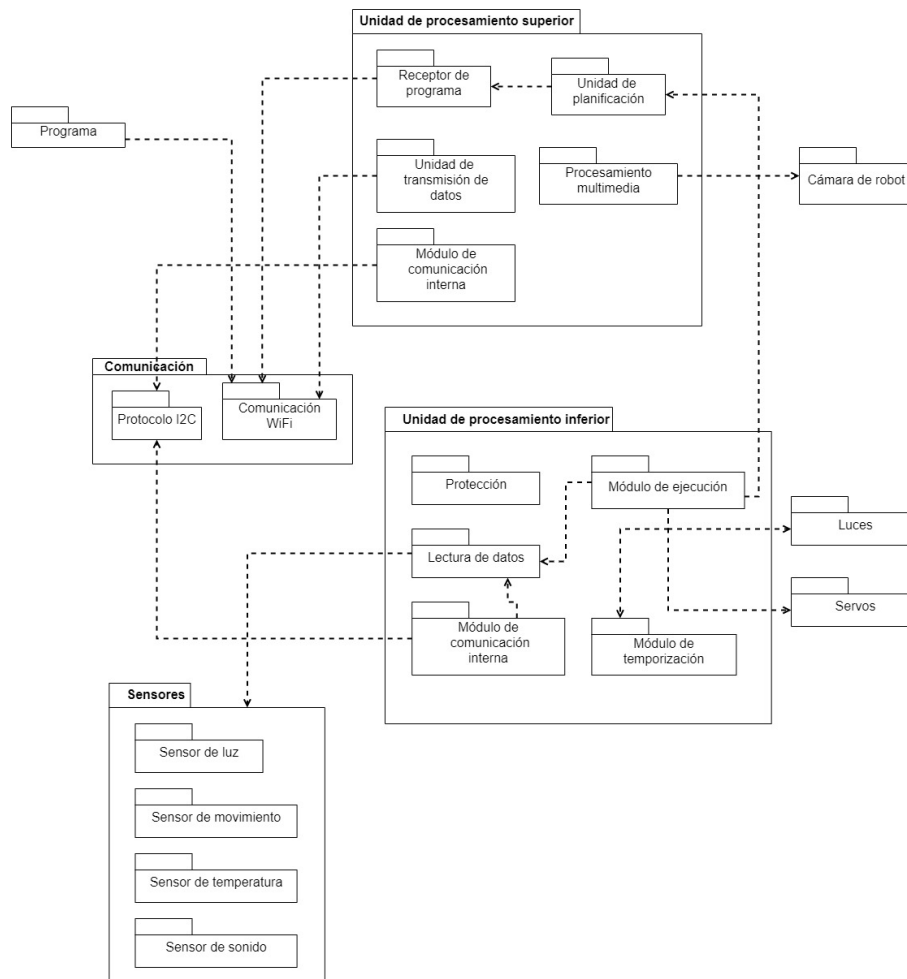


Figura 3: Diagrama de paquetes del sistema.

## 6.5. Información

No aplica.

## 6.6. Uso de patrones

Para este proyecto, el patrón de diseño de fábrica abstracta se aplica en el nivel superior de funcionalidad. Específicamente, se utilizará al procesar las instrucciones que se deben ejecutar en el robot. De modo que para cada una de las acciones se utilice un solo código para ejecutar la lógica interna de la instrucción en lugar de crear código especializado para cada una de las instrucciones, esto produciría un código poco escalable, en caso de que se agreguen más comandos en el futuro. El diagrama de clases que se propone se muestra en la figura 4.

En caso de que se necesiten agregar más instrucciones, solo se necesita crear la clase que maneja dicha instrucción y hacerla heredar de la clase abstracta. Además de implementar las acciones que ejecutan cuando se necesita activar la instrucción.

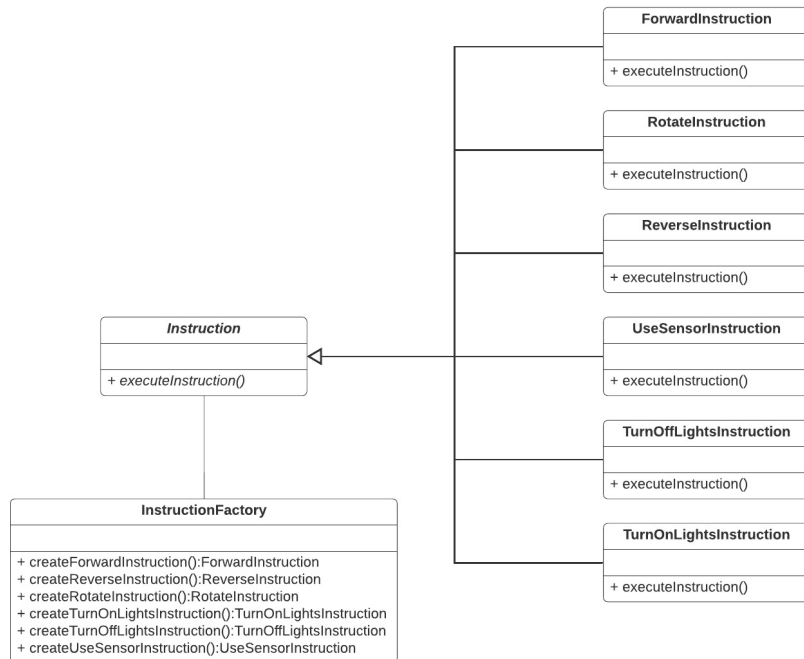


Figura 4: Diagrama de clases para el patrón de diseño de fábrica abstracta para el procesamiento de instrucciones de los programas.

## 6.7. Interfaces

La única interfaz que ofrece el sistema diseñado es la comunicación inalámbrica que se utiliza para la transmisión de datos de vídeo y fotos o la recepción del programa por ejecutar. Por lo tanto, si se requiere comunicar con el producto, el único requisito es la conexión a internet inalámbrica, como lo ejemplifica el diagrama de la figura 5.

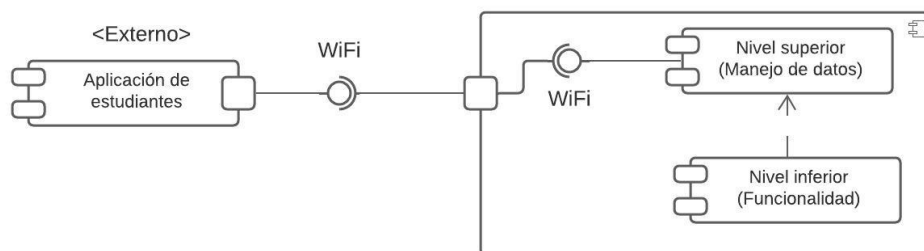


Figura 5: Diagrama de interfaces de comunicación para el diseño de la solución.

## 6.8. Interfaces de usuario

No aplica.

## 6.9. Estructura

Para el diagrama de estructura de la figura 6 se especifican los componentes estructurales básicos de la arquitectura. Para la parte inferior, se tienen todos los módulos de medición, luces, temporizadores y servos. En el nivel superior los bloques principales son el módulo de comunicación inalámbrica y la cámara del robot.

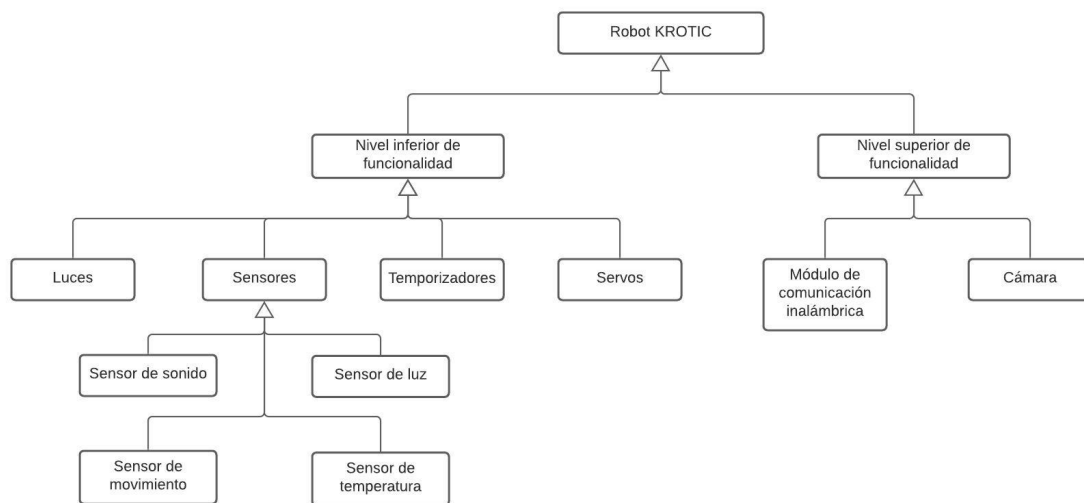


Figura 6: Diagrama de estructura del sistema.

## 6.10. Interacción

En el diagrama de secuencia de la figura 7 se observa que el proceso inicia cuando se detecta un programa proveniente de los estudiantes. Primero, el nivel superior de funcionalidad tomará el programa y planificará las acciones exactas a realizar. El nivel superior empezará a enviar las instrucciones de forma serial y al mismo tiempo activará la cámara para documentar y transmitir las acciones que están siendo ejecutadas por el robot.

Luego, el nivel inferior de funcionalidad activará los módulos de *hardware* que sean requeridos por las instrucciones. Si los sensores son utilizados, los valores medidos son reportados de vuelta al nivel superior para ser transmitidos. El flujo finaliza cuando el programa es ejecutado por completo.

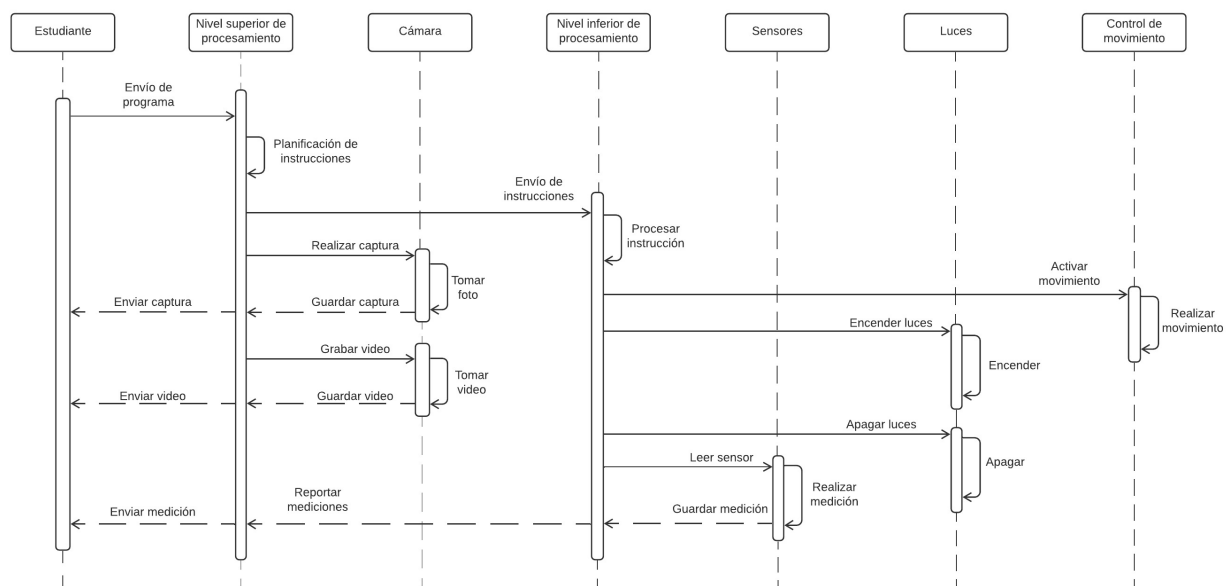


Figura 7: Diagrama de secuencia para la ejecución de un programa en la arquitectura.

## 6.11. Dinámica de estados

No se documenta.

## 6.12. Algoritmos

No aplica.

## 6.13. Recursos

No se documenta.

## 7. Apéndices

### 7.1. Apéndice A. Alternativas de diseño

Para cada una de las alternativas de diseño, los factores que se consideraron más críticos son: consumo energético, tamaño y peso. En la tabla 4, se incluyen otros factores de importancia como capacidad de procesamiento y soporte de entradas.

Para los aspectos de tamaño y peso, se refieren a las limitaciones físicas del robot. El robot es un cubo de quince centímetros (15 cm) de lado, por lo tanto es necesario considerar el tamaño de los componentes que se vayan a utilizar. Para el peso se debe considerar que el conjunto de componentes no dificulte el movimiento con los motores que actualmente se tienen.

El consumo energético parte de la base de que se quiere que el robot sea amigable con el ambiente. Asimismo, el tamaño de la batería que se defina también influye en el peso y espacio de ocupación. Por lo que se quiere que sea la menor posible para realizar todas las funciones.

#### 7.1.1. Propuesta de dos niveles simple

Para la propuesta de la figura 8, se tiene la solución más básica del conjunto. Esta consta de únicamente dos niveles, donde el nivel superior de funcionalidad es el encargado de manejar los datos percibidos por la cámara y de la transmisión de video y fotos hacia el estudiante. El nivel inferior se encargará del manejo de sensores, temporizadores, luces y movimiento del robot. La comunicación entre ambos niveles es a través de I<sup>2</sup>C, que se utiliza para intercambio de instrucciones, así como reporte de mediciones de los sensores.

El convertidor de analógico a digital se utiliza para leer los valores de nivel de batería y pasárselos a la Raspberry Pi, de modo que se utilicen las luces indicadores, de modo que se enciendan cuando alguna de las baterías se encuentra cerca de dejar de ser útil para que pueda ser recargada o reemplazada.

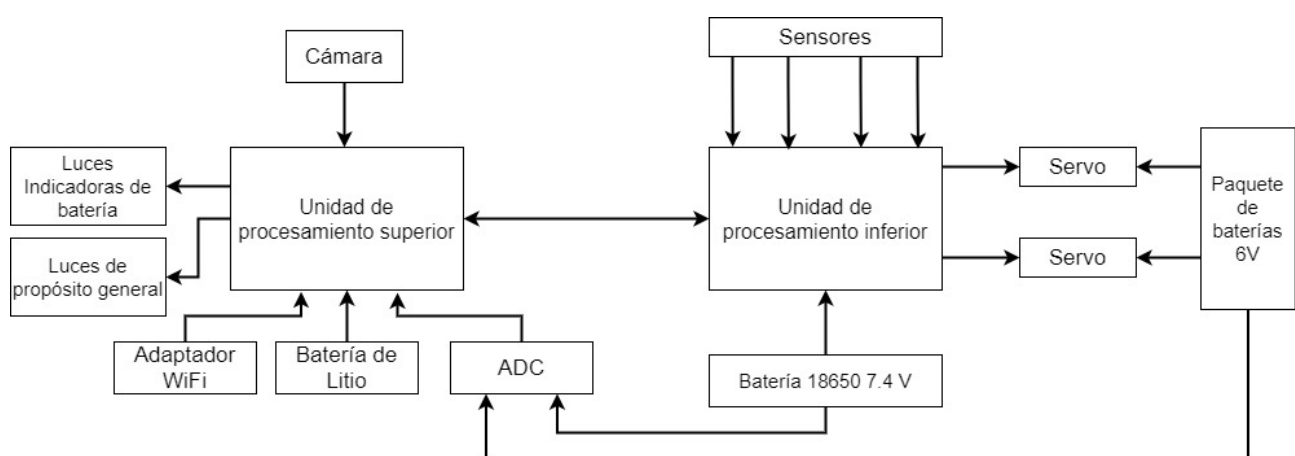


Figura 8: Diagrama de bloques para la propuesta de dos niveles simple.

### 7.1.2. Propuesta de dos niveles con nivel inferior separado

En la solución de la figura 9, se observa que en este caso, el nivel inferior de funcionalidad se divide con el propósito de especializar las unidades que componen el nivel. Por un lado, una unidad de procesamiento se encarga de la mayor parte de control de sensores y uso de temporizador y por otro lado, la unidad secundaria se encarga del movimiento, luces y del sensor de luz.

A diferencia de la primera alternativa, el nivel inferior no se encuentra tan saturado de componentes. Sin embargo, el consumo energético se ve incrementado, el espacio requerido también aumenta, así como el peso. Aún así, la capacidad de procesamiento y soporte de dispositivos aumenta significativamente.

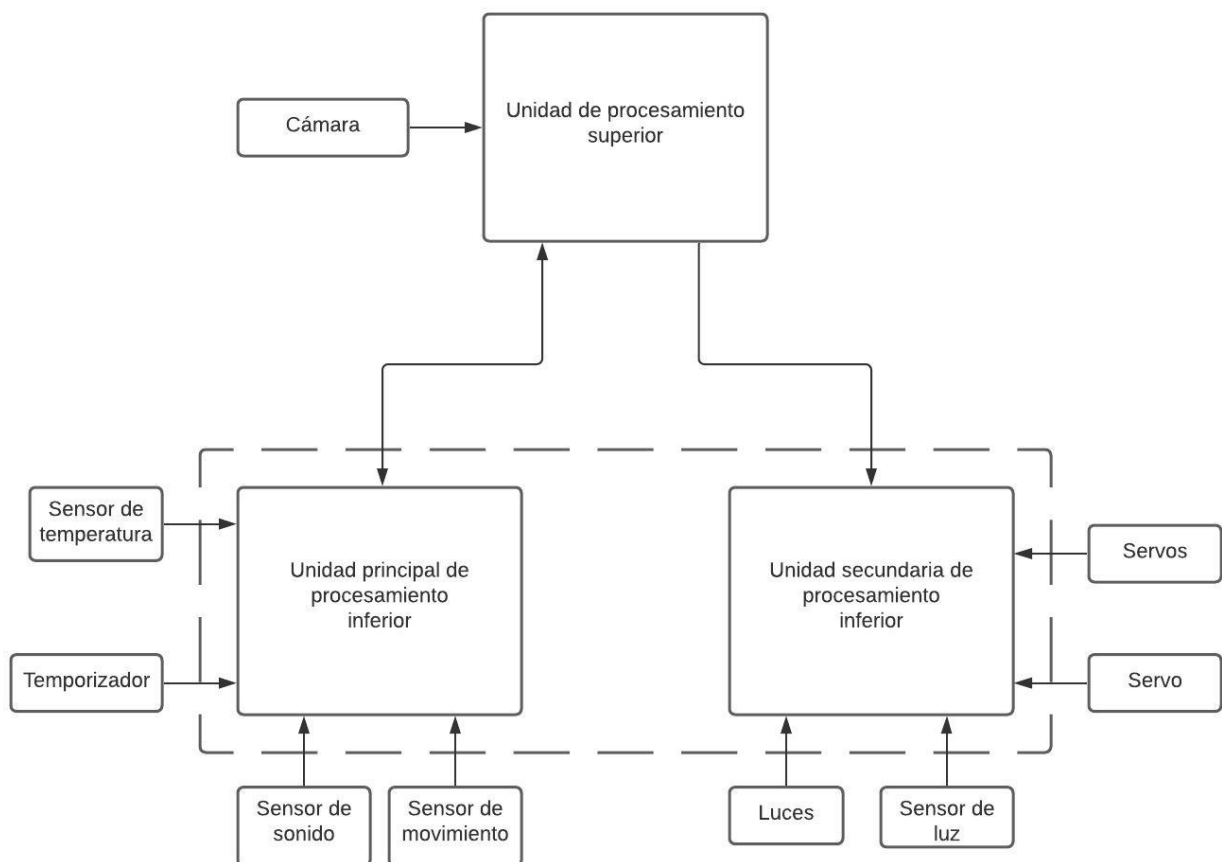


Figura 9: Diagrama de bloques para la propuesta de dos niveles con separación del nivel inferior.

### 7.1.3. Propuesta de dos niveles con unidad intermediaria de control

Para la tercera propuesta se introduce una unidad intermediaria entre los niveles. Esta unidad funciona como centro de control para las funcionalidades de todos los otros niveles. Es el encargado de recibir los programas a ser ejecutados y con base a eso, se comunican

las operaciones respectivas que debe realizar cada unidad. Es el punto intermediario de intercambio de datos entre los niveles.

También, se piensa que si en un futuro se agregan  $n$  cantidad de niveles adicionales, esta unidad de control puede reducir la complejidad de comunicación entre los niveles. De modo que se tenga un solo canal de comunicación centralizado en lugar de tener  $n$  canales.

Similar a la segunda propuesta, al agregar una unidad, el consumo se ve afectado negativamente, igual como el peso aumenta. Una desventaja que posee cuando se compara con la segunda propuesta es que no aumenta la cantidad de entradas soportadas, ya que el propósito de la unidad no es manejar componentes externos como sensores, sino servir como canal de comunicación centralizado.

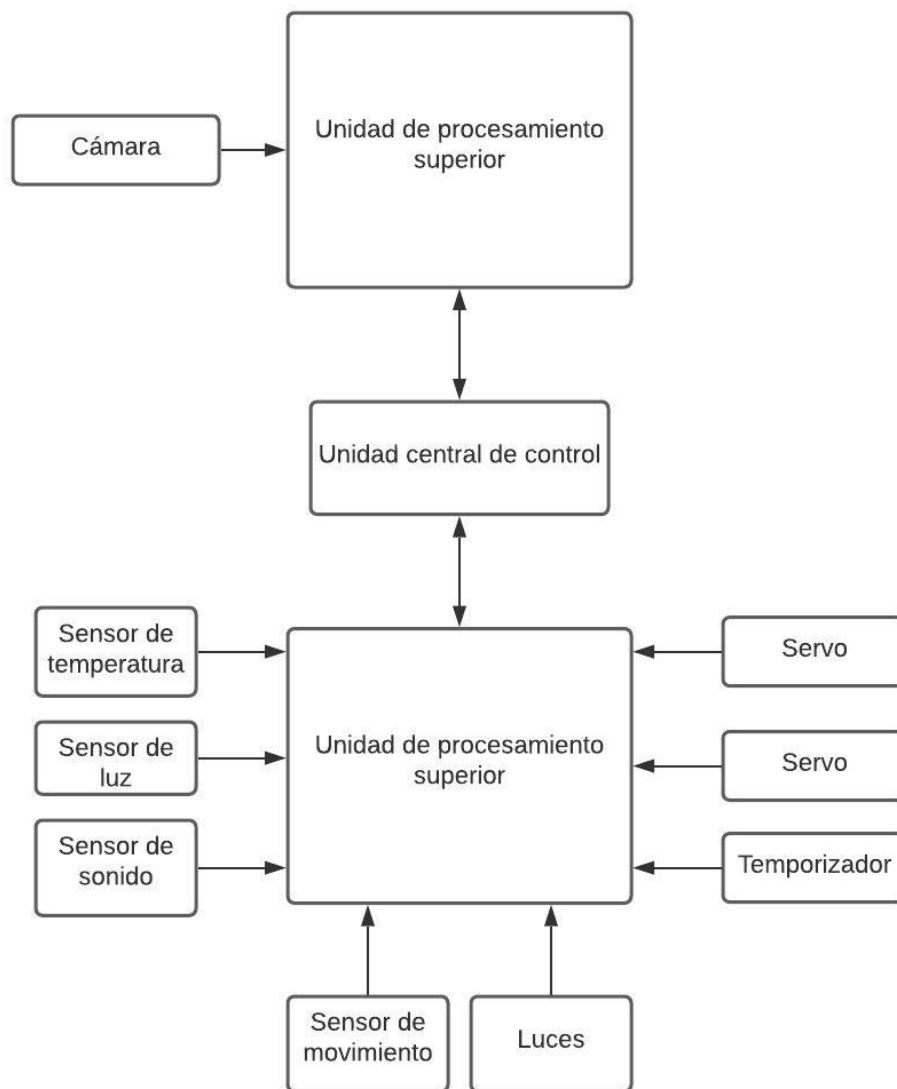


Figura 10: Diagrama de bloques para la propuesta de dos niveles utilizando una unidad intermediaria de control

## 7.2. Apéndice B. Comparación de propuestas de diseño

Cuadro 4: Tabla de comparación de propuestas

Propuesta	Consumo	Tamaño	Peso	Soporte de entradas	Procesamiento
Simple de dos niveles.	Moderado.	Moderado.	El peso no limita las funcionalidades de movimiento.	Intermedio. Depende de las tarjetas de desarrollo.	Intermedio. Se divide únicamente entre dos niveles.
Nivel inferior dividido.	Incrementa con respecto a la propuesta simple.	Incrementa con respecto a la propuesta simple.	Incrementa con respecto a la propuesta simple. Sin limitar las funcionalidades de movimiento significativamente.	Aumenta significativamente con respecto a la propuesta simple.	Aumenta significativamente con respecto a la propuesta simple.
Unidad intermedia.	Incrementa con respecto a la propuesta simple.	Incrementa con respecto a la propuesta simple.	Incrementa ligeramente con respecto a la propuesta simple, de modo que no limita el movimiento del robot.	No cambia con respecto a la propuesta simple.	Aumenta ligeramente con respecto a la propuesta simple.

### 7.2.1. Selección de propuesta

Basándose en los datos de la tabla 4, los diseños que se presentan favorables son: la propuesta simple de dos niveles y la propuesta de dos niveles con nivel inferior dividido. La propuesta con unidad intermedia no presenta muchos beneficios con respecto a las otras dos propuestas que puedan ser de inmediata utilidad, como se discutió en la sección 7.1.3, la utilidad de esta propuesta vendría a ser considerable si existiera una mayor cantidad de unidades.



Entonces, dadas las restricciones de espacio físico principalmente, que por lo tanto afecta la autonomía del robot, se ha elegido la opción simple de dos niveles para implementar en el prototipo. Para este prototipo se utilizará una Raspberry Pi para el nivel superior y un Arduino UNO para el nivel inferior ya que poseen las herramientas más convenientes para implementar el prototipo y además son los componentes con los que se tiene mayor documentación externa, por lo tanto, la ubicación de información será fácil incluso para aquellos que manejen la arquitectura por primera vez.

En cuanto a implementación, referirse al apéndice C, donde se muestra una tabla comparativa con las posibles opciones para las tarjetas de desarrollo que serán las unidades principales para cada nivel. En la tabla se muestran otros atributos de interés para el sistema, como la alimentación, memoria, cantidad de pines, entre otros.

### 7.3. Apéndice C. Tabla de tarjetas de desarrollo candidatas para implementación

Cuadro 5: Características de tarjetas de desarrollo candidatas para la implementación.

<b>Dispositivo</b>	<b>Alimentación</b>	<b>Memoria</b>	<b>Puertos totales</b>	<b>Precio aproximado</b>
Raspberry Pi 3 B+	2.5A/5V	1 GB	40	\$40
Raspberry Pi 4	3A/5V	2/4/8 GB	40	\$52
BeagleBone Black	2A/5V	512 MB	69	\$35-\$80
ASUS Tinker Board S	3A/5V	2 GB	28	\$95
Arduino Mega	7-12V	8 KB	54	\$40
Rock64	3A/5V	4 GB	40	\$45
Odroid-XU4	4A/5V	2 GB	38	\$100