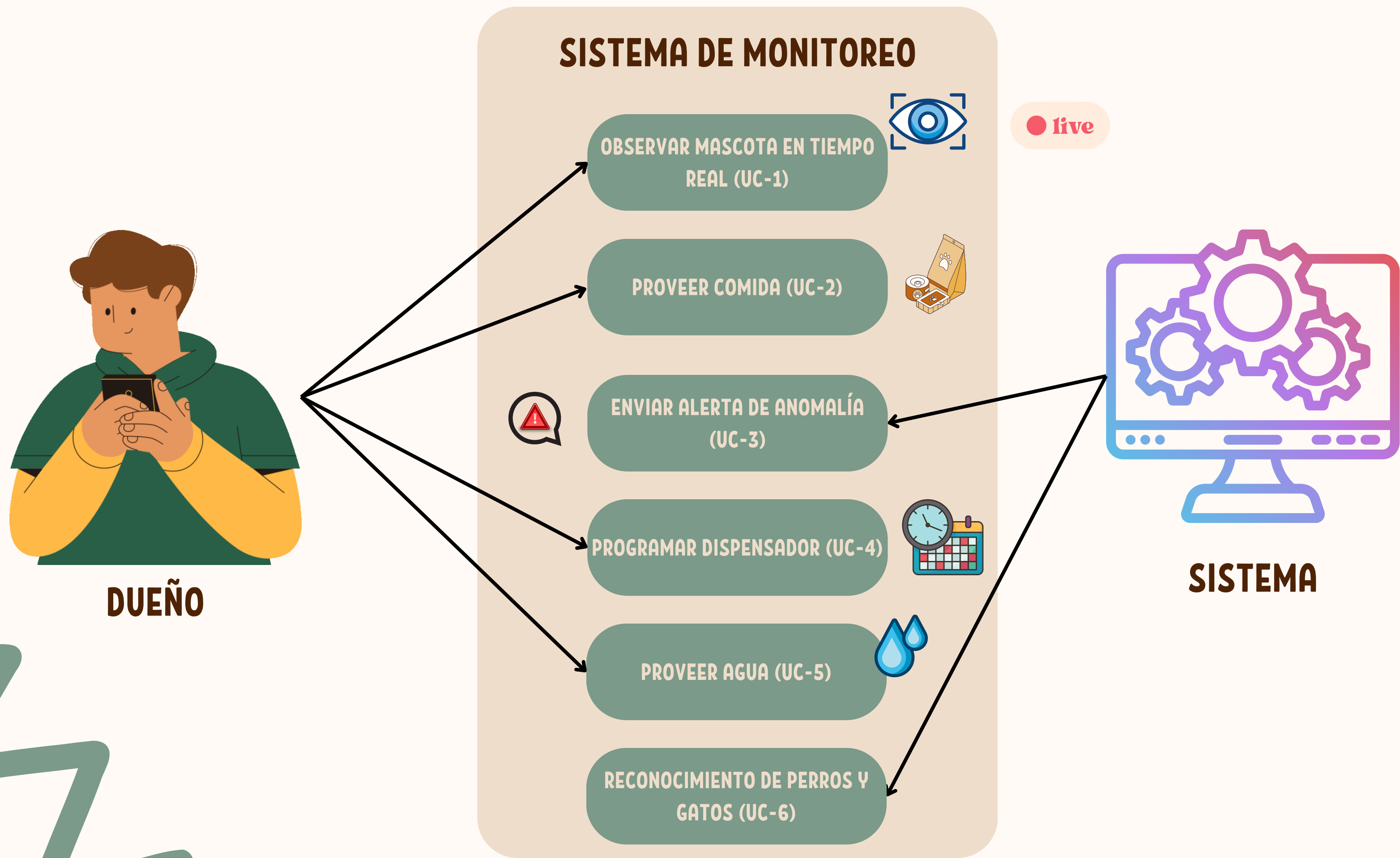


# CASOS DE USO QUE IMPACTAN EN LA ARQUITECTURA

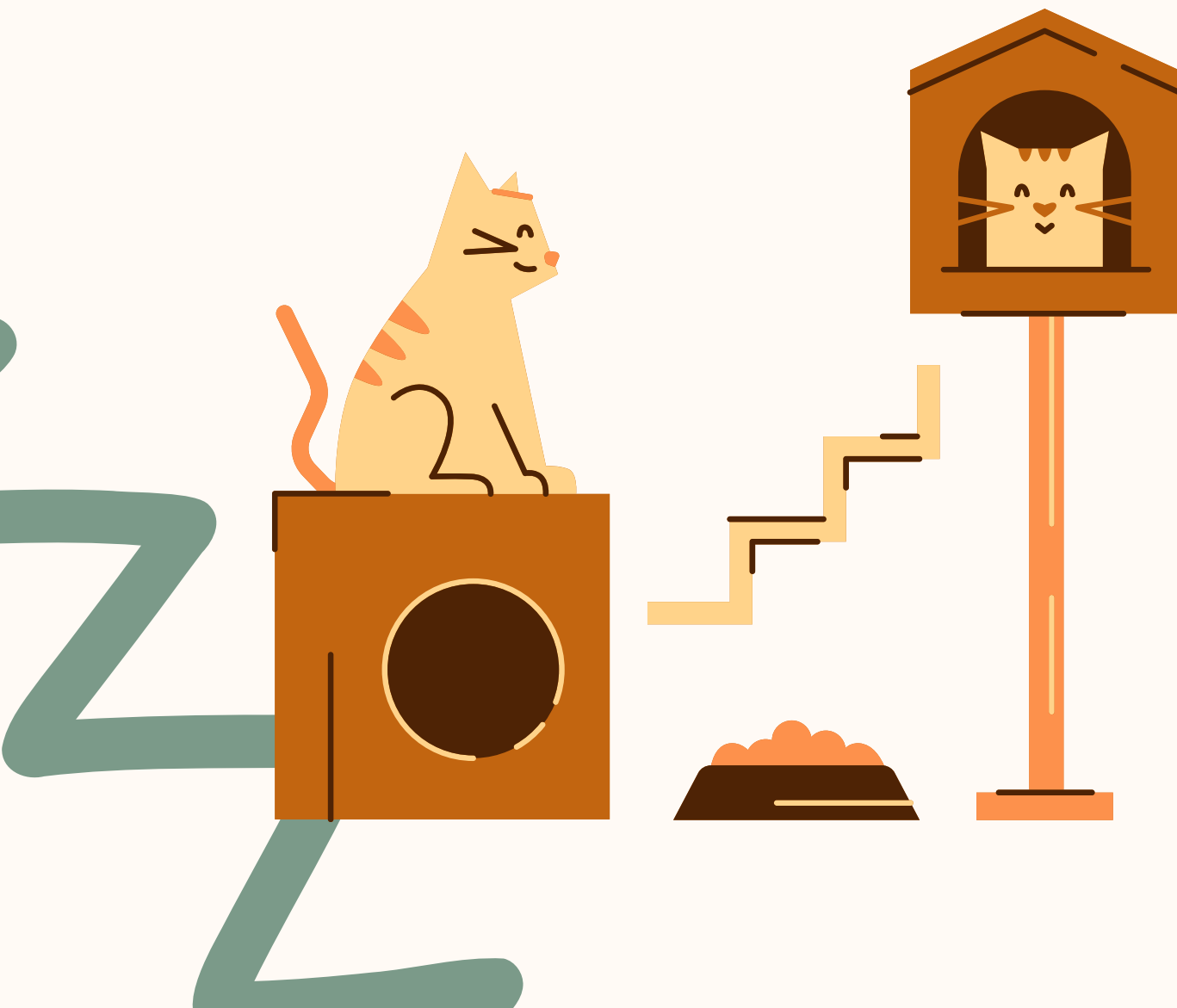


# ATRIBUTOS DE CALIDAD

**USABILIDAD**

**RENDIMIENTO**

**SEGURIDAD**



# ESCENARIOS - ATRIBUTOS ①

## USABILIDAD - QA1



**FUENTE**

**DUEÑO DE LA MASCOTA**

**ESTÍMULO**

**INTERACCION CON EL SISTEMA**

**ARTEFACTO(S)**

**APLICATIVO MÓVIL Y PÁGINA WEB**

**AMBIENTE**

**HOGAR DEL USUARIO Y SU  
MASCOTA**

**RESPUESTA**

**PERMITE AL USUARIO REALIZAR LA TAREA DESEADA  
SIN COMPLICACIONES**

**MEDICIÓN DE  
RESPUESTA**

**LAS VALORACIONES DE SATISFACCIÓN DEL  
USUARIO Y EL PORCENTAJE DE TAREAS  
COMPLETADAS SATISFACTORIAMENTE DE  
97%**

# ESCENARIOS - ATRIBUTOS ②

## RENDIMIENTO- QA2



**FUENTE**

**DUEÑO DE LA MASCOTA**

**ESTÍMULO**

**SELECCIONAR CÁMARA EN VIVO**

**ARTEFACTO(S)**

**CÁMARAS CON CONEXIÓN A  
INTERNET**

**AMBIENTE**

**HOGAR DEL CLIENTE Y SU  
MASCOTA**

**RESPUESTA**

**MOSTRAR CÁMARA EN VIVO**

**MEDICIÓN DE  
RESPUESTA**

**$T \leq 5 \text{ Seg}$**

# ESCENARIOS - ATRIBUTOS ③

## SEGURIDAD - QA3

FUENTE

PERSONA EXTERNA

ESTÍMULO

INTENTO DE ACCESO NO  
AUTORIZADO

ARTEFACTO(S)

SISTEMA DE AUTENTICACIÓN

AMBIENTE

SISTEMA DE PET MONITORING

RESPUESTA

EL SISTEMA BLOQUEA EL ACCESO Y  
ENVÍA UNA ALERTA AL DUEÑO DE LA  
MASCOTA

MEDICIÓN DE  
RESPUESTA

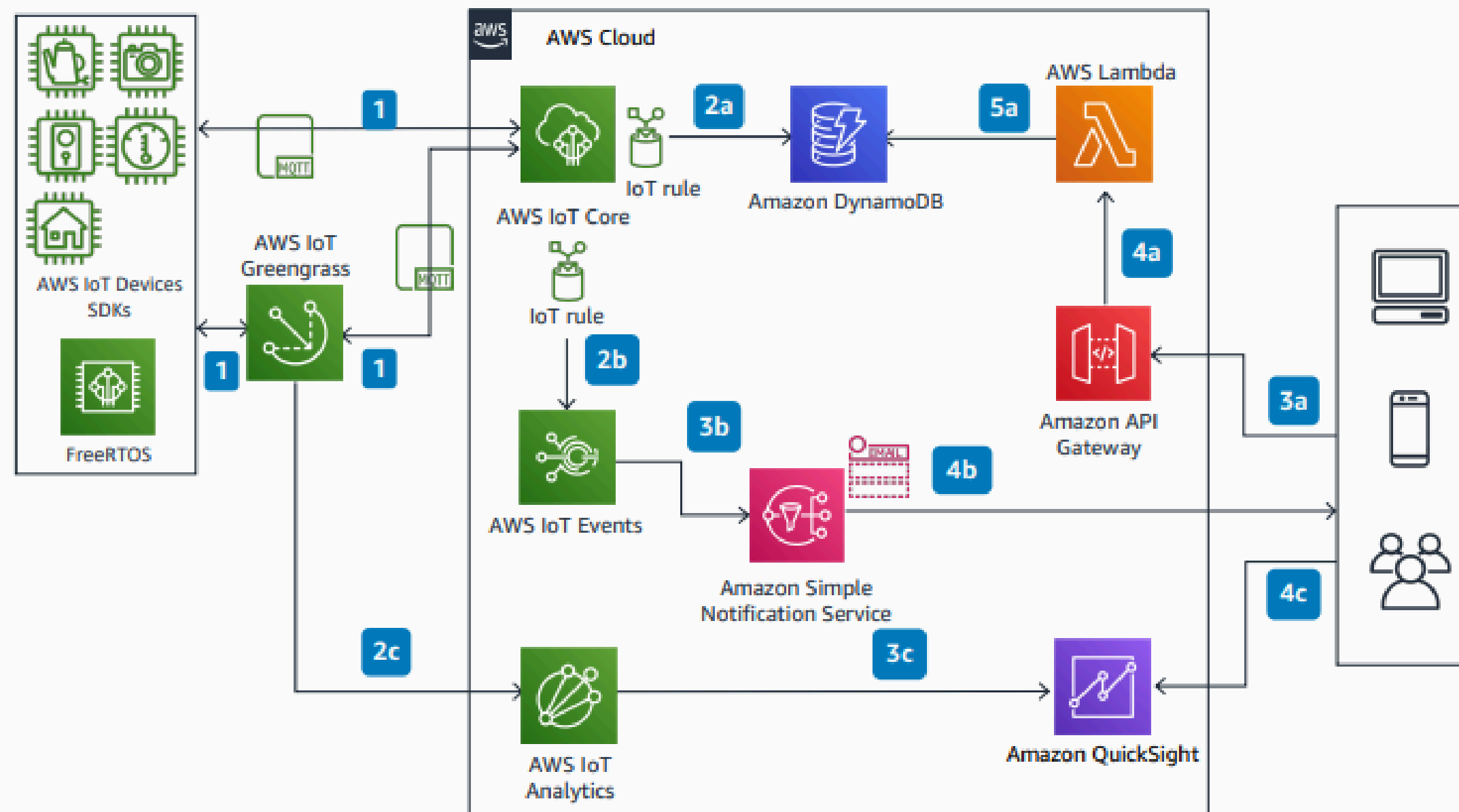
$T \leq 3 \text{ Seg}$



# ARQUITECTURA DE REFERENCIA 1 - (AR1)

## Connected Home - Telemetry

Measuring and collecting data from smart home devices



Reviewed for technical accuracy March 22, 2021  
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

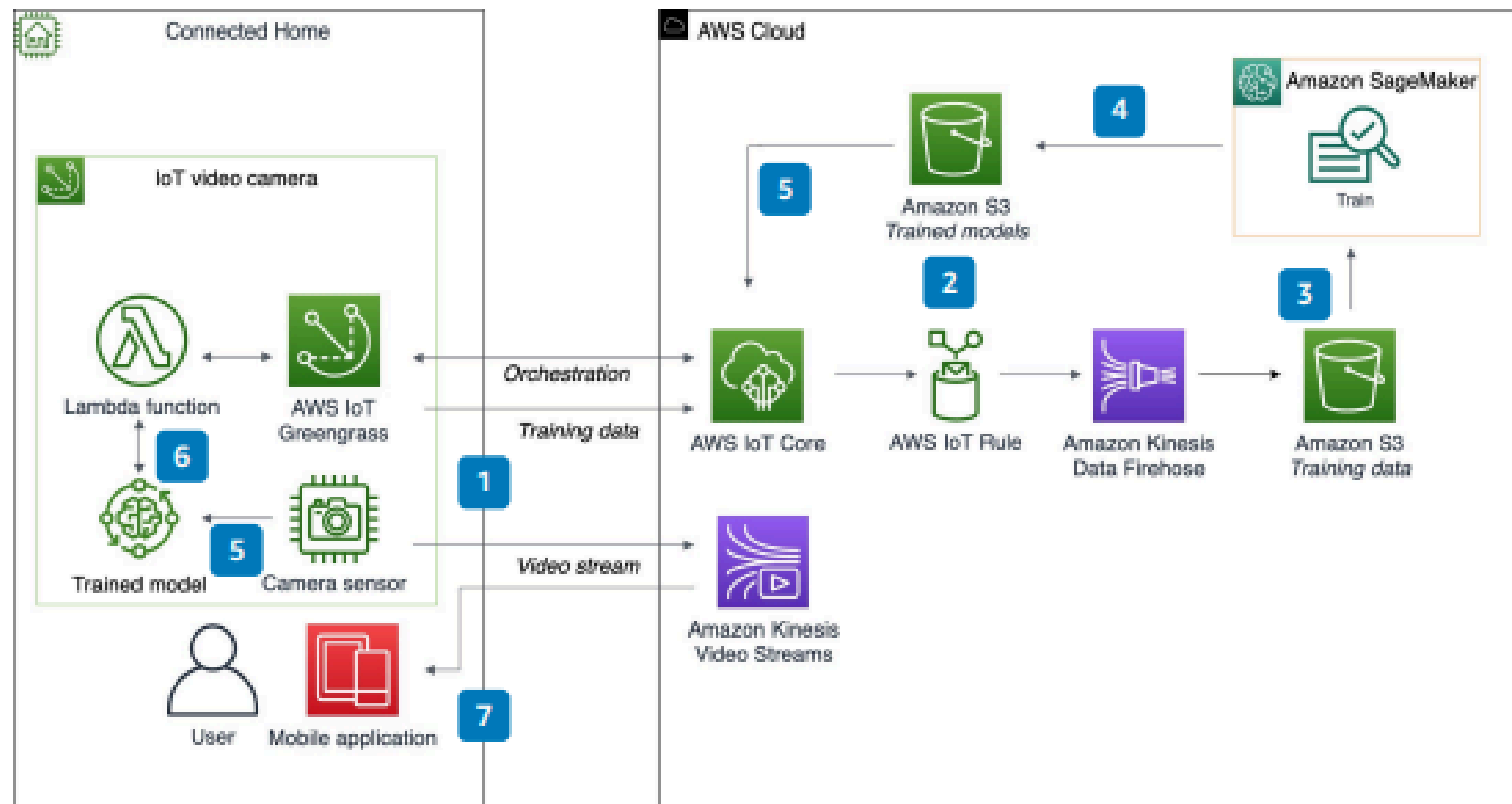
AWS Reference Architecture

- 1** The devices connect to **AWS IoT Core**, directly via MQTT from **FreeRTOS** or via **AWS IoT Greengrass**.
- 2a** An IoT topic Rule can be configured in **AWS IoT Core** to pick up meaningful data and save it in **Amazon DynamoDB**.
- 2b** Another IoT topic Rule can be configured in **AWS IoT Core** to trigger **AWS IoT Events** for certain data, like temperature etc.
- 2c** Or, directly from **AWS IoT Greengrass**, data can be ingested in **AWS IoT Analytics**.
- 3a** Client applications can make an API call to retrieve data via **Amazon API Gateway**.
- 3b** **AWS IoT Events** can trigger **Amazon SNS** actions in case certain changes occur in the data (e.g. Temperature > 25 degrees).
- 3c** Data from **AWS IoT Analytics** can be imported into **Amazon QuickSight**, for visualization and further insights.
- 4a** **Amazon API Gateway** invokes an **AWS Lambda** function with logic to retrieve data requested by the client applications from **Amazon DynamoDB**.
- 4b** **Amazon SNS** triggers an email notification for client applications.
- 4c** Client applications can visualize the data in **Amazon QuickSight**.

# ARQUITECTURA DE REFERENCIA 2 - (AR2)

## Connected Home – Machine Learning at the Edge

IoT Machine Learning on Home Devices



- 1 An IoT video camera running **AWS IoT Greengrass** publishes training data to **AWS IoT Core**.
- 2 An **IoT Rule** that is listening for camera data forwards messages to **Amazon Kinesis Data Firehose** for storage in **Amazon S3**.
- 3 **Amazon SageMaker** is used to train, optimize, and build machine learning (ML) models that can use less than a tenth of the memory footprint found in resource-constrained devices like cameras.
- 4 **Amazon SageMaker** outputs trained models to **Amazon S3** for delivery to the IoT video camera.
- 5 **AWS IoT Greengrass** cloud service is used to orchestrate deployments of software to the target IoT video camera, including trained models, and application logic such as an **AWS Lambda** function or Docker container.
- 6 A **Lambda** function running locally on the IoT video camera performs inference using the latest version of the trained ML model.
- 7 The homeowner can use mobile applications to view the camera's video stream via **Amazon Kinesis Video Streams**.



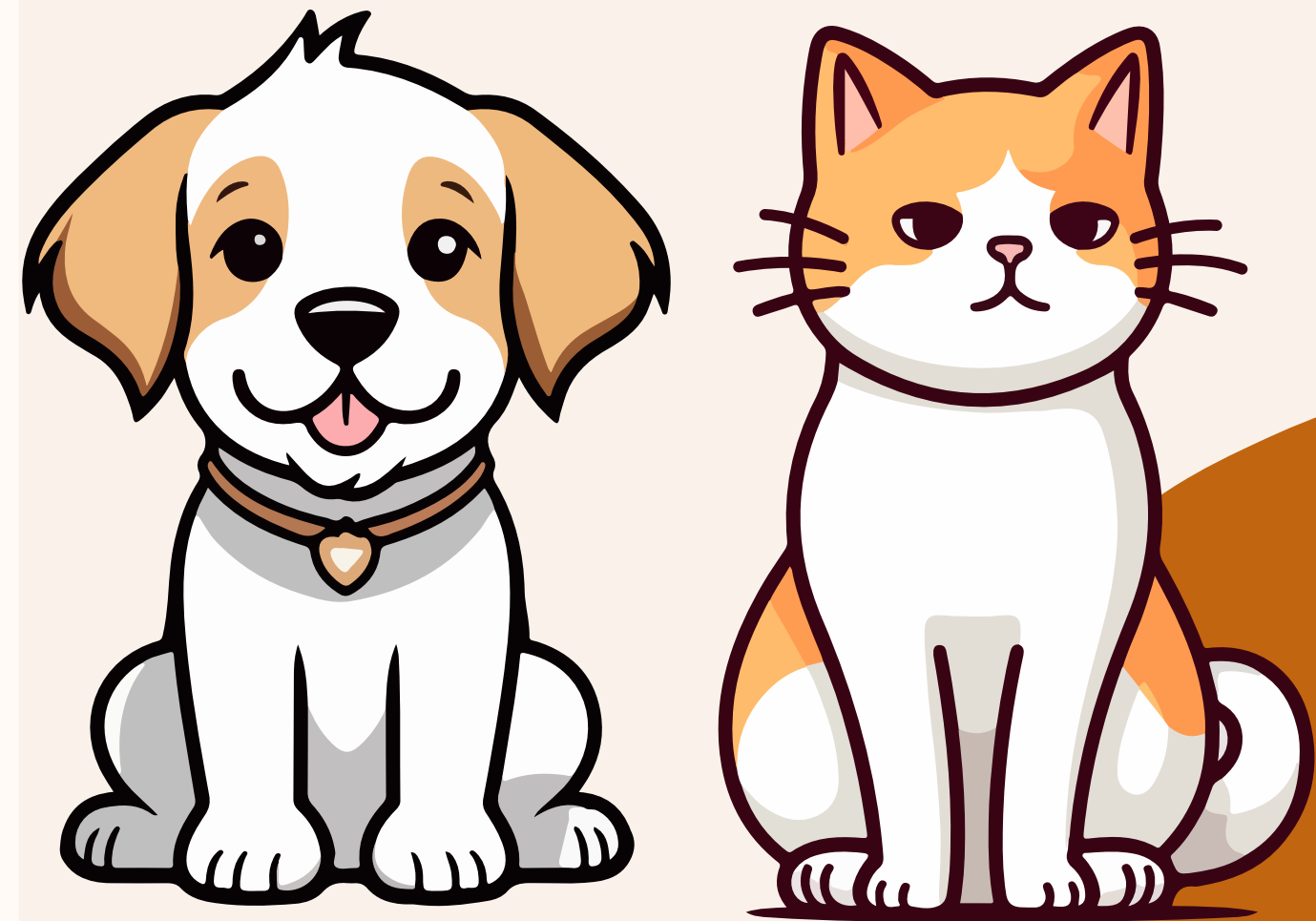
Reviewed for technical accuracy July 13, 2021  
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Reference Architecture

## ATTRIBUTE DRIVEN DESIGN 3.0

- Paso 1: Revisar las entradas
- Paso 2: Establecer el objetivo de la iteración por medio de la selección de drivers
- Paso 3: Escoger uno o más elementos del sistema para ser refinados
- Paso 4: Escoger uno o más conceptos de diseño para satisfacer los drivers seleccionados
- Paso 5: Instanciar los elementos arquitecturales, asignar responsabilidades y definir interfaces
- Paso 6: Boceto de las vistas y registro de las decisiones de diseño
- Paso 7: Ejecutar el análisis del diseño actual, revisar el objetivo de la iteración y logros del diseño

## PROCESO ADD ARQUITECTURA





# ITERACIÓN 1:

## PASO 1: REVISAR LAS ENTRADAS

Propósito del diseño	Implementar una solución de monitoreo para mascotas en el hogar
Requerimientos funcionales primarios	UC1-UC2-UC3-UC4-UC5-UC6
Escenarios de atributos de calidad	QA1-QA2-QA3
Restricciones	CON1-CON2-CON3-CON4
Arquitecturas existentes	AR1-AR2

## PASO 2: ESTABLECER EL OBJETIVO DE LA ITERACIÓN POR MEDIO DE LA SELECCIÓN DE DRIVERS

- Objetivo de la iteración: Desarrollar y desplegar el sistema que cumpla con los casos de uso (UC\*), atributos de calidad (QA\*) y con las restricciones (CON\*).

## PASO 3: ESCOGER UNO O MÁS ELEMENTOS DEL SISTEMA PARA SER REFINADOS

Modificar las arquitecturas de referencia para cumplir con los drivers del paso 2.

## PASO 4: ESCOGER UNO O MÁS CONCEPTOS DE DISEÑO PARA SATISFACER LOS DRIVERS SELECCIONADOS

Decisiones de diseño y alojamiento	Rationale
Arquitectura de microservicios	Alta flexibilidad, escalabilidad y mantenibilidad del sistema.
Implementación de una arquitectura de autenticación segura	Garantiza que solo usuarios autorizados puedan acceder al sistema.
Uso de una arquitectura IoT para cámaras con conexión a Internet	Permite el monitoreo en tiempo real de las mascotas desde cualquier lugar.
Control de dispensadores mediante arquitectura IoT	Proporciona la capacidad de alimentar a las mascotas de forma remota y programada.

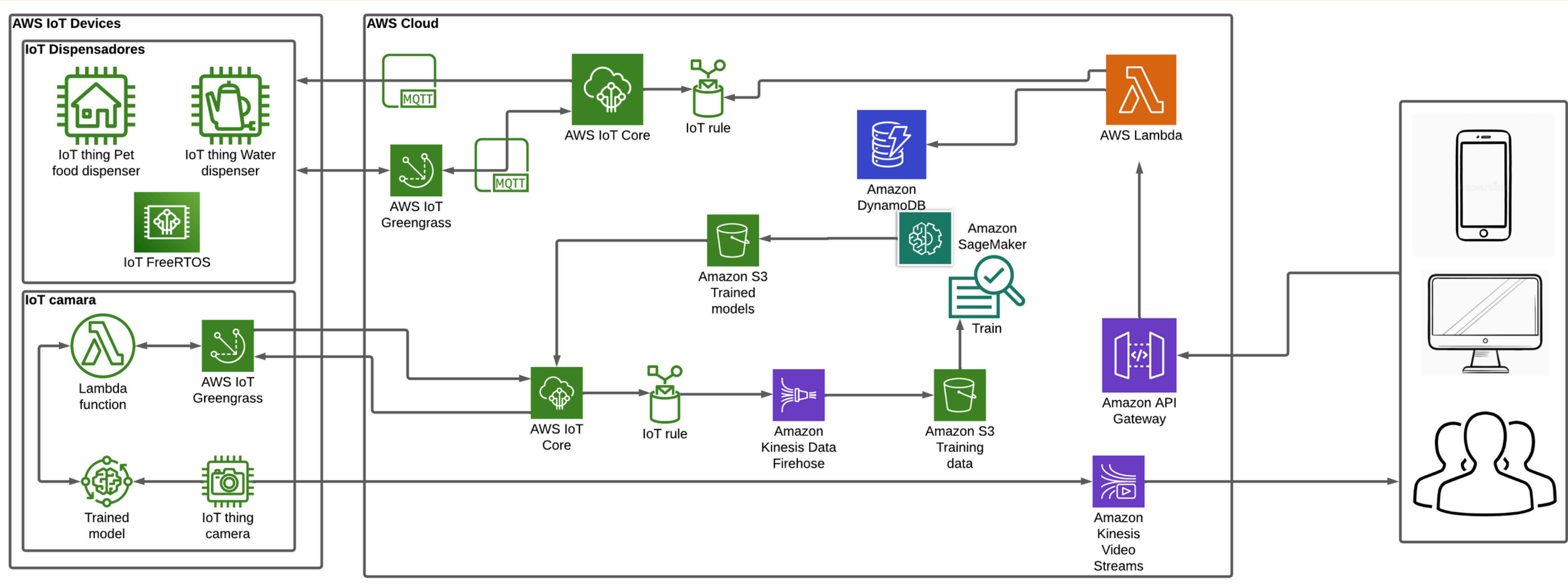
## PASO 5: INSTANCIAR LOS ELEMENTOS ARQUITECTURALES, ASIGNAR RESPONSABILIDADES Y DEFINIR INTERFACES

Decisiones de diseño y alojamiento	Rationale
IoT Dispensadores	Control de dispensadores de comida y agua mediante IoT FreeRTOS. Permite alimentar a las mascotas remotamente, asegurando que siempre tengan acceso a comida y agua fresca. Esto mejora la satisfacción del dueño al poder controlar la alimentación de sus mascotas desde cualquier lugar.
IoT Cámara	Captura de video y monitoreo en tiempo real con procesamiento de imágenes mediante modelos entrenados. Esto asegura que los dueños puedan monitorear la actividad de sus mascotas, mejorando la seguridad y tranquilidad al saber que sus mascotas están bien cuidadas.
AWS IoT Greengrass	Procesamiento local de datos y comunicación con AWS IoT Core. Facilita la gestión eficiente de los datos generados por los dispositivos IoT, optimizando el uso de la red y reduciendo la latencia.

PASO 5:

Decisiones de diseño y alojamiento	Rationale
AWS IoT Core	Recolectar y gestionar datos de los dispositivos IoT. Centraliza la gestión de los datos de IoT, mejorando la integridad y disponibilidad de la información.
Amazon DynamoDB	Almacenamiento inmutable y seguro de datos de monitoreo y eventos. Garantiza la integridad y seguridad de los datos almacenados, protegiéndolos contra modificaciones no autorizadas.
AWS Lambda	Ejecución de lógica de procesamiento según los datos recibidos. Permite una respuesta rápida a eventos en tiempo real, mejorando la eficiencia del sistema.
Amazon SageMaker	Entrenamiento y despliegue de modelos de machine learning para procesamiento de imágenes y eventos. Mejora la capacidad de análisis y respuesta del sistema mediante técnicas avanzadas de aprendizaje automático.
Amazon API Gateway	Proveer una interfaz para la comunicación entre aplicaciones y servicios. Facilita la integración de los diversos componentes del sistema, asegurando una comunicación fluida y segura.
Amazon Kinesis Data Firehose	Ingesta y procesamiento de datos en tiempo real para análisis y almacenamiento. Asegura que los datos se procesen y almacenen de manera eficiente y en tiempo real, mejorando la capacidad de análisis del sistema.
Amazon Kinesis Video Streams	Transmisión de video en tiempo real para visualización y análisis. Permite a los dueños ver el video en tiempo real de sus mascotas, asegurando una vigilancia continua.

# PASO 6: BOCETO DE LAS VISTAS Y REGISTRO DE LAS DECISIONES DE DISEÑO



# PASO 7: EJECUTAR EL ANÁLISIS DEL DISEÑO ACTUAL, REVISAR EL OBJETIVO DE LA ITERACIÓN Y LOGROS DEL DISEÑO

No realizado	Parcialmente realizado	Completamente realizado	Decisiones de diseño realizados durante la iteración
-	-	UC1 - QA2	Monitoreo en tiempo real de las cámaras: Uso de Amazon Kinesis para procesamiento y análisis de datos en tiempo real.
-	-	UC6	Reconocimiento de perros y gatos: Implementación de Amazon SageMaker para entrenamiento y despliegue de modelos.
-	-	CON3 - CON4	Conectar la app móvil/página web a los dispensadores: Uso de IoT FreeRTOS para dispositivos de control de dispensadores. Integración de AWS IoT Core para gestión y comunicación de dispositivos IoT. Uso de AWS Lambda para enviar órdenes a esos dispositivos.
-	-	CON2	Debido a que se cuenta con un sistema de monitoreo y reconocimiento con las cámaras del hogar, ya se tiene el espacio del hogar establecido de la vigilancia.
-	CON1	-	La función Lambda solo se encarga de dispensar la comida cuando se le envía la orden, mas aún no se puede programar a una hora en específico.
-	-	UC2 - UC5	Mediante API, Lambda, IoT Core se pueden controlar los dispensadores para realizar la dispensación de comida y bebida en el momento que se selecciona la opción.
UC3- UC4- QA3	-	-	-
-	QA1	-	Actualmente, con las funciones y microservicios implementados, no se ha cumplido con todos los casos de uso, por lo que la usabilidad sigue siendo un punto a reforzar.



# ITERACIÓN 2:

## PASO 2: ESTABLECER EL OBJETIVO DE LA ITERACIÓN POR MEDIO DE LA SELECCIÓN DE DRIVERS

- Objetivo de la iteración: Desarrollar el sistema de alertas necesario para el sistema y agregar la función para programar dispensación para cumplir con lo faltante en la iteración previa.

## PASO 3: ESCOGER UNO O MÁS ELEMENTOS DEL SISTEMA PARA SER REFINADOS

- Sistema de alertas.
- Sistema de notificaciones de órdenes completadas.
- Función para programar dispensación.
- Sistema de autenticación de usuario.

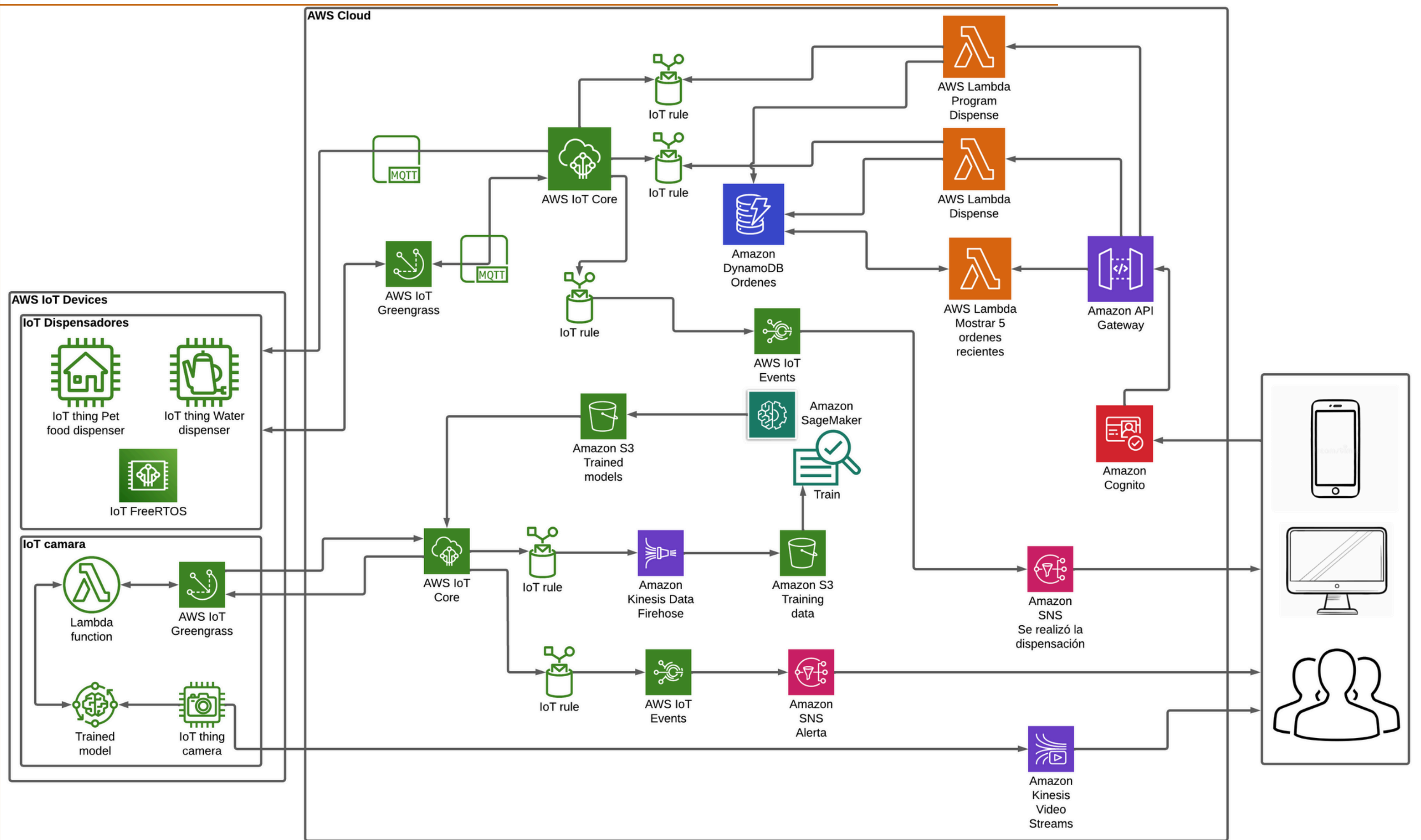
## PASO 4: ESCOGER UNO O MÁS CONCEPTOS DE DISEÑO PARA SATISFACER LOS DRIVERS SELECCIONADOS

Decisiones de diseño y alojamiento	Rationale
Implementación de alertas (SNS) con eventos (IoT Events)	Brinda una mayor usabilidad al sistema porque la interacción usuario - sistema será más eficiente.
Uso de Amazon Cognito para autenticación	Asegura que solo usuarios autorizados puedan acceder al sistema, mejorando la seguridad y la gestión de identidades.
Implementación de más funciones Lambda	Esto nos permitirá satisfacer el resto de casos de uso faltantes en la 1ra iteración

# PASO 5: INSTANCIAR LOS ELEMENTOS ARQUITECTURALES, ASIGNAR RESPONSABILIDADES Y DEFINIR INTERFACES

Decisiones de diseño y alojamiento	Rationale
AWS SNS	Implementa el sistema de notificaciones para alertar a los usuarios sobre eventos importantes u anomalías, mejorando la comunicación y la interacción con el usuario.
Amazon Cognito	Implementa la autenticación y autorización de usuarios para mejorar la seguridad del sistema, asegurando que solo usuarios legítimos puedan acceder a la plataforma.
AWS Lambda Program Dispense	Esta función permite que el usuario pueda programar una hora y fecha a la cual desee dispensar a su mascota.
AWS Lambda Mostrar 5 órdenes recientes	Esta función permite al usuario observar sus 5 órdenes más recientes. De esta forma puede asegurarse de que su orden ha sido registrada.

# PASO 6: BOCETO DE LAS VISTAS Y REGISTRO DE LAS DECISIONES DE DISEÑO





# PASO 7: EJECUTAR EL ANÁLISIS DEL DISEÑO ACTUAL, REVISAR EL OBJETIVO DE LA ITERACIÓN Y LOGROS DEL DISEÑO

No realizado	Parcialment e realizado	Completamente realizado	Decisiones de diseño realizados durante la iteración
-	-	UC3	Implementación de un sistema de alertas que notifica al usuario en caso de no detectar a la mascota del usuario, utilizando AWS SNS.
		UC4	Implementación de una nueva función AWS Lambda que permite programar la fecha y hora de la dispensación de agua o comida.
-	-	QA3	Implementación de un sistema de autenticación segura para la gestión de usuarios, asegurando que solo personas autorizadas accedan al sistema usando AWS Cognito.
-	-	QA1	Desarrollo de una funcionalidad que permite a los usuarios visualizar las órdenes completadas recientemente, mejorando la usabilidad del sistema. Además, se agregó la nueva función de programar dispensación, lo cual es otra herramienta útil para el usuario.
-	-	CON1	El desarrollo de este sistema permite dispensar comida o agua en el momento, programar la dispensación, observar a la mascota a través de las cámaras, recibir notificaciones de alarma y de que se realizó la dispensación de forma adecuada. Esto en conjunto resulta en una función completa que permite al usuario cumplir de forma remota con los cuidados básicos de sus mascotas.

# CONCLUSIONES

- SE IDENTIFICARON ARQUITECTURAS DE REFERENCIA QUE FUERON UTILES PARA LA CRACIÓN DEL DIAGRAMA DE ARQUITECTURA PARA LAS ITERACIONES DEL MÉTODO ADD.
- ES UNA BUENA FORMA DE PODER AYUDAR A LOS DUEÑOS DE LAS MASCOTAS A CONTROLAR LO QUE HACEN MIENTRAS SE ENCUENTRAN FUERA DE CASA.
- EL DESARROLLO DE UN SISTEMA DE MONITOREO PARA MASCOTAS NO SOLO ELEVA EL NIVEL DE ATENCIÓN Y PROTECCIÓN DE LAS MISMAS, SINO QUE TAMBIÉN MARCA UN PROGRESO IMPORTANTE EN LA INCORPORACIÓN DE TECNOLOGÍA EN LA VIDA DIARIA.

