

# Sistema de visualización de datos de sensores

Documento de Requisitos de Usuario

Working Draft 0.4  
30-ENE-2019

Introducción .....	2
Propósito .....	2
Ámbito del sistema .....	2
Definiciones y acrónimos .....	2
Referencias .....	3
Resumen del documento .....	3
Descripción general .....	4
Perspectiva de producto .....	4
Características generales .....	5
Restricciones generales .....	5
Características de usuario .....	6
Suposiciones y dependencias .....	6
Entorno operacional .....	7
Requisitos específicos .....	8
Capacidades generales .....	8
Otras restricciones .....	10
Apéndices .....	12
Posible formato de configuración .....	12
Detalles sobre los procesos modulares .....	13

# Introducción

Esta sección incluye un resumen a grandes rasgos de la estructura de este documento.

Este documento incluye la documentación de requisitos de usuario para el sistema a desarrollar, conocido actualmente como **Sistema de Visualización de Datos de Sensores**. La estructura del documento sigue el estándar PSS-05-0 de la ESA sobre Estándares de Desarrollo de Software.

## Propósito

Finalidad de este documento, audiencia objetivo y otras aclaraciones.

El principal propósito de este documento es definir de manera inequívoca las funcionalidades, limitaciones y alcance del sistema a desarrollar. Para ello, está destinado a su uso por parte de todos los agentes interesados, directa e indirectamente en el mismo.

## Ámbito del sistema

Visión general del sistema en desarrollo.

El sistema a desarrollar se basa principalmente en ofrecer una agregación de datos producidos por sensores remotos de forma que los usuarios puedan consultarlos desde una única plataforma. La comunicación con los sensores se hará a través de un sistema lo más modular posible, para facilitar la conexión de nuevos sensores.

## Definiciones y acrónimos

Glosario de términos técnicos y otros acrónimos que puedan aparecer en el documento.

- **.NET Standard:** Especificación de .NET disponible en todas las implementaciones, con independencia de la plataforma.
- **.NET Core:** Es una implementación de .NET Standard para uso general, modular, multiplataforma y de código abierto.
- **ASP.NET Core:** Framework multiplataforma para crear aplicaciones conectadas a internet, que se ejecuta sobre .NET Core.
- **Web API:** Framework para construir servicios HTTP que puedan ser consumidos por muchos clientes, basado en peticiones REST.

- **REST:** Interfaz entre sistemas que utiliza HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos.

## Referencias

Conjunto de recursos útiles a los que se puede hacer referencia en este documento.

- ESA: Software engineering and standardisation - PSS
  - [http://www.esa.int/TEC/Software\\_engineering\\_and\\_standardisation/TECBUCUXBQE\\_0.html](http://www.esa.int/TEC/Software_engineering_and_standardisation/TECBUCUXBQE_0.html)
- Guía de .NET Core | Microsoft Docs
  - <https://docs.microsoft.com/es-es/dotnet/core/index>
- Introduction to ASP.NET Core | Microsoft Docs
  - <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2>
- Build Web APIs with ASP.NET Core | Microsoft Docs
  - <https://docs.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-2.2>

## Resumen del documento

Describe qué información está incluida y dónde se puede encontrar.

Tras la introducción del apartado anterior, que debería dar una idea básica del sistema, se detallan a continuación aspectos más específicos que centran las características del sistema y otras restricciones iniciales.

Más adelante aparece el listado de requisitos (funcionales y no funcionales) del sistema, lo que constituye la parte más técnica del documento.

En último lugar aparecerán apéndices con información adicional, como diagramas u otros aspectos, según se estime necesario.

# Descripción general

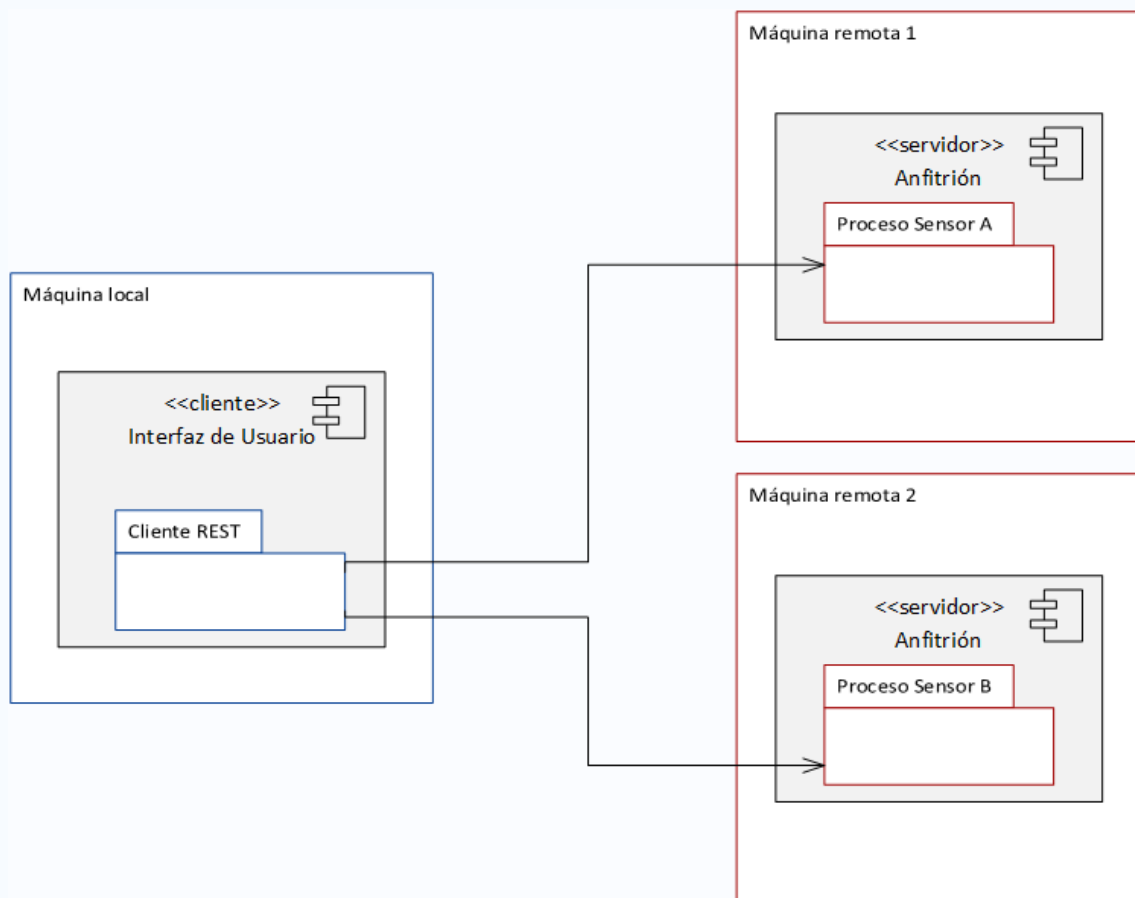
Aspectos técnicos y otras suposiciones sobre el sistema.

## Perspectiva de producto

El sistema a desarrollar deberá comunicarse con un número indeterminado y variable de sensores, previamente instalados y en funcionamiento en el momento de la conexión. Estos sensores pueden estar instalados en máquinas remotas, con las que el sistema se deberá comunicar.

Tanto el número como los ajustes propios de cada sensor podrán ser configurados por el usuario mediante la interfaz de usuario del sistema.

Las conexiones con los sensores utilizarán un esqueleto común como anfitrión (encargado de gestionar mensajes de usuario y enviar datos), y un proceso modular que se ejecutará dentro del anfitrión (encargado de conseguir los datos de cada sensor). De esta forma, para conectar un nuevo sensor basta con desplegar el sistema anfitrión común en la máquina remota y acoplar el proceso apropiado para ese sensor:



En principio es posible que en una misma máquina haya más de un sensor, siempre que se ejecuten en la misma tantos procesos modulares como sensores conectados. Si se da este caso, la dirección IP y el puerto serán comunes a todos los sensores.

Además, tanto el cliente como los procesos modulares necesitarán almacenar configuraciones y datos intermedios o temporales en archivos de configuración, que se guardarán en la máquina local (para el cliente) o en las máquinas remotas (para los procesos modulares).

## Características generales

Entre las características del sistema de cara a sus usuarios finales se pueden destacar las siguientes:

- Permitir a los usuarios **recibir datos de varios sensores** en una misma interfaz, aunque no necesariamente de forma simultánea, sino permitiendo filtrar los sensores que se muestran según criterios significativos.
- Permitir a los usuarios **modificar el número de sensores conectados** al sistema, pudiendo añadir nuevos sensores, eliminar alguno de los existentes o modificar su configuración (cuando sea posible) desde la interfaz del sistema.
- Permitir a los usuarios **ver y modificar la configuración** del sistema y de la interfaz de usuario.
- Facilitar el uso del sistema a los usuarios, de forma que **no necesiten conocimientos técnicos o internos** del sistema.

## Restricciones generales

El sistema se adaptará a los tipos de sensor que sean considerados durante el desarrollo del mismo. Aunque la arquitectura será intencionadamente flexible para facilitar la integración de nuevos sensores, no se garantiza que estos puedan ser conectados al sistema con posterioridad, si resultan ser incompatibles con la misma.

Más específicamente, el proceso de conectar un sensor se puede dividir en dos pasos: Obtener los datos, y mostrarlos al usuario. La obtención de datos puede solucionarse creando un nuevo proceso modular apropiado. Por otro lado, la visualización de datos por parte del usuario puede ser problemática si dichos datos tienen un formato diferente al previsto.

La integración de tipos de sensor no compatibles, si se estima necesaria, requerirá modificaciones al sistema que no están cubiertas en este documento.

Como restricción específica del cliente, el desarrollo del sistema (y de sus dependencias, cuando sea posible) se llevará a cabo sobre la plataforma **.NET**, usando el lenguaje de programación **C#**.

La arquitectura de comunicaciones prevista implica desplegar software que actúe como servidor en las máquinas remotas con sensores. El núcleo del sistema y la interfaz de usuario actuarán como clientes emitiendo peticiones a dichos servidores.

## Características de usuario

Aunque el sistema podrá ser usado por cualquier usuario final, los siguientes perfiles de usuario se tendrán en cuenta de forma específica durante el desarrollo:

- **Usuarios no técnicos:** Estos usuarios no tienen necesariamente conocimientos específicos sobre el sistema o sus dependencias, y su uso del mismo se limita a la visualización de los datos agregados, ejecutando únicamente operaciones sencillas sobre los mismos, como filtrados o similar.
- **Usuarios técnicos informáticos:** Estos usuarios tienen conocimientos extensos de informática, y pueden usar el sistema en profundidad para modificar la configuración del mismo, utilizar toda su funcionalidad y solucionar algunos problemas que puedan encontrarse, como errores de configuración o permisos de acceso.
- **Usuarios técnicos no informáticos:** Estos usuarios tienen conocimientos técnicos, pero no necesariamente de informática, sino seguramente sobre los sensores o la infraestructura de comunicaciones. Su uso del sistema estará por tanto centrado en el proceso de conexión y mantenimiento de sensores, donde podrán diagnosticar y solucionar fallos relacionados con su especialidad, como parámetros específicos del hardware o de la red.

Como nota a este apartado, el sistema no contempla la gestión o autenticación de cuentas de usuario. Los perfiles mencionados solo se tendrán en cuenta para el diseño de los aspectos no funcionales del sistema.

## Suposiciones y dependencias

El sistema a desarrollar parte de las siguientes suposiciones para su funcionamiento:

- Los sensores que se conectarán al sistema estarán **integrados en máquinas dedicadas**. En cualquier caso, dichas máquinas no ejecutarán procesos pesados o erráticos que puedan afectar al rendimiento de la misma de forma imprevista.
- Tanto los sensores como las máquinas remotas a las que están conectados **estarán en línea** y disponibles **de forma más o menos continua** (salvo casos

excepcionales como mantenimiento o avería), sin depender de intervención humana u horarios restrictivos.

- Si alguno de los sensores fuera desconectado de forma externa, el sistema se limitará a informar de esa circunstancia mediante mensajes de error que se mostrarán a los usuarios locales.
- Será necesario **desplegar software propio** del sistema **en las máquinas remotas** con el fin de establecer conexiones y transmitir datos desde los sensores a la interfaz de usuario. Por ello, dicho software deberá permanecer alojado y en ejecución (potencialmente de forma continua) en las máquinas remotas para que el sistema funcione correctamente.
  - En algunos casos, el software propio podría configurarse para que se **inicie de forma automática** al encender la máquina remota en la que está alojado.
- Adicionalmente, los procesos modulares desplegados podrían recibir ciertas actualizaciones o modificaciones de forma remota, que serán aplicadas y puestas en ejecución de forma autónoma por el esqueleto anfitrión. Si el alcance de las modificaciones hiciera imposible su transmisión automática o se tuviera que modificar el propio esqueleto anfitrión, el software deberá ser desplegado manualmente de nuevo.
- Si algún sensor o máquina remota está incluido en **una infraestructura de red especial** por motivos de seguridad o confidencialidad, puede que sea necesario contactar con los responsables antes de conectar dicho sensor al sistema, con el fin de que permitan la comunicación necesaria entre las interfaces de usuario y el anfitrión desplegado.
  - El diseño actual necesita que se habilite una dirección IP y un puerto accesibles desde el exterior.
- Por su parte, el sistema estará disponible para su uso potencialmente de forma continua (salvo casos excepcionales como mantenimiento o avería), aunque no tenga ningún sensor conectado o datos para mostrar.
- La máquina en la que se despliegue la interfaz de usuario del sistema deberá ser dedicada, o en cualquier caso no ejecutar procesos pesados o erráticos que puedan afectar al rendimiento de forma imprevista.

## Entorno operacional

A falta de una documentación de arquitectura más específica, el sistema necesitará comunicarse con las siguientes interfaces externas:

- **Módulos anfitrión remotos:** Componentes software propios que se desplegarán en las máquinas remotas para actuar como servidores Web API



y transmitir datos recogidos en dichas máquinas en respuesta a peticiones de los clientes.

- **Procesos modulares:** Desplegados en las máquinas remotas, dentro de los anfitriones. Estos procesos se usarán para comunicarse con los sensores o realizar tareas de mantenimiento y diagnóstico del sistema o la infraestructura de comunicaciones, y podrían ejecutarse de forma continua, periódica y/o a petición.
- **Interfaz de usuario:** Aplicación o servicio software diseñado para ejecutarse de forma local y consumir y/o modificar los datos de procesos modulares, con fines de consulta, mantenimiento o demostración.

## Requisitos específicos

Incluye un listado de los requisitos que este sistema deberá cumplir.

## Capacidades generales

### Requisitos para usuarios

RF.1: Los usuarios podrán acceder al sistema de forma anónima, sin necesidad de registrarse o identificarse.

RF.2: Cada vez que accedan al sistema, este mostrará una selección predefinida de datos.

RF.3: El sistema permitirá a los usuarios aplicar filtros para ver solo un subconjunto de los sensores conectados.

RF.3.1: El hecho de aplicar más de un filtro se interpretará como una operación lógica AND. Es decir, solamente los datos que cumplan todos los criterios activos serán mostrados.

RF.3.2: El sistema ofrecerá los siguientes criterios de filtrado:

RF.3.2.1: Nombre del sensor (coincidencia total o parcial, no sensible a mayúsculas).

RF.3.2.2: Tipo del sensor

RF.4: El sistema permitirá al usuario eliminar filtros previamente aplicados, de uno en uno.

RF.5: El sistema permitirá al usuario eliminar todos los filtros aplicados a la vez.

RF.6: El usuario deberá confirmar su selección de filtros para que la vista de datos se actualice.

### Requisitos para sensores

RF.7: Los usuarios podrán añadir nuevos sensores al sistema.

- RF.7.1: Estos sensores deberán utilizar un método de comunicación compatible con el sistema.
- RF.7.2: Para ello, los usuarios deberán indicar los siguientes parámetros de conexión:
- RF.7.2.1: Dirección IP
  - RF.7.2.2: Puerto
  - RF.7.2.3: Tipo de sensor
- RF.7.3: El usuario escribirá un nombre identificativo para el nuevo sensor, que podrá contener caracteres alfanuméricos y espacios.
- RF.7.3.1: Ese nombre será usado siempre que el sistema deba referirse al sensor de forma visible para el usuario.
  - RF.7.3.2: El nombre será único entre todos los sensores conectados.
- RF.7.4: Cada tipo de sensor podrá ofrecer una serie de valores de configuración propios como parte del proceso de conexión.
- RF.7.4.1: Aunque estos ajustes propios sean incorrectos, el sistema completará el proceso de conexión, y se limitará a indicar que no es posible recibir datos de ese sensor.
- RF.8: Los usuarios podrán eliminar sensores existentes del sistema.
- RF.8.1: Este proceso será irreversible, de forma que, si el usuario quiere añadir el sensor de nuevo, deberá hacerlo desde cero.
  - RF.8.2: Sin embargo, el esqueleto anfitrión y el proceso modular de la máquina remota no serán alterados por este proceso, sino que la eliminación actúa solamente en el cliente.
- RF.9: Los usuarios podrán modificar los siguientes parámetros de un sensor actualmente conectado al sistema:
- RF.9.1: Dirección IP
  - RF.9.2: Puerto
  - RF.9.3: Tipo de sensor
  - RF.9.4: Cada uno de sus valores de configuración propios.
- RF.10: El sistema permitirá validar el formato de los parámetros de conexión indicados en los requisitos RF.7.2 y RF.9 antes de guardarlos.
- RF.10.1: Aunque el formato supere la validación, el sistema no garantizará que el contenido en sí sea correcto o tenga sentido.
- RF.11: El sistema será compatible con los siguientes tipos de sensores:
- RF.11.1: Sensor láser lineal Gocator, de LMI Technologies.

## Requisitos para configuración

- RF.12: El sistema ofrecerá la posibilidad de que los usuarios modifiquen valores de configuración global o de la interfaz de usuario.
- RF.12.1: Estos valores persistirán entre ejecuciones del sistema.

RF.13: El sistema permitirá al usuario validar el formato de los valores de configuración que lo admitan antes de guardarlos.

RF.13.1: Aunque el formato supere la validación, el sistema no garantizará que el contenido en sí sea correcto o tenga sentido.

RF.14: El sistema permitirá restaurar los valores por defecto de la configuración global y de la interfaz de usuario.

## Otras restricciones

RNF.1: La máquina en la que vaya a desplegarse la interfaz de usuario tendrá instalada una plataforma compatible.

~~RNF.1.1: Si la interfaz es una aplicación de escritorio WPF, la plataforma necesaria será .NET Framework 4.6.1 o superior.~~

RNF.1.2: La interfaz de usuario es una aplicación UWP, y la plataforma necesaria será Windows 10 Versión 1803 o superior.

RNF.2: Las máquinas remotas en las que se vayan a desplegar servidores anfitrión tendrán instalada la plataforma .NET Core 3

RNF.3: Tanto la máquina local con interfaz de usuario como las máquinas remotas con sensores tendrán conexión a internet.

RNF.4: Los usuarios no técnicos deberán ser capaces de usar el sistema de forma básica tras un máximo de dos horas de aprendizaje o entrenamiento.

RNF.4.1: El uso básico del sistema comprende la siguiente funcionalidad:

RNF.4.1.1: Abrir o acceder a la interfaz de usuario.

RNF.4.1.2: Visualizar los datos que se muestran por defecto.

RNF.4.1.3: Aplicar un filtro para modificar los sensores mostrados.

RNF.4.1.4: Eliminar el filtro para volver a la vista inicial.

RNF.4.2: Dicho aprendizaje o entrenamiento podrá consistir en una lectura autónoma de la documentación o en un aprendizaje guiado por otra(s) persona(s).

RNF.5: Los usuarios técnicos informáticos deberán ser capaces de usar el sistema de forma extendida tras un máximo de seis horas de aprendizaje o entrenamiento.

RNF.5.1: El uso extendido del sistema comprende la siguiente funcionalidad:

RNF.5.1.1: Abrir o acceder a la interfaz del sistema.

RNF.5.1.2: Visualizar los datos que se muestran por defecto.

RNF.5.1.3: Aplicar uno o varios filtros para modificar los datos mostrados.

RNF.5.1.4: Eliminar los filtros para volver a la vista inicial.

RNF.5.1.5: Ver un listado de los problemas actuales, o la ausencia de estos.

RNF.5.1.6: Acceder a un listado de los sensores actualmente conectados.

RNF.5.1.7: Ver el estado actual de cada uno de esos sensores.

RNF.5.1.8: Añadir un nuevo sensor compatible al sistema.

- RNF.5.1.9: Eliminar un sensor conectado del sistema.
- RNF.5.1.10: Acceder a la configuración del sistema.
- RNF.5.2: Dicho aprendizaje o entrenamiento podrá consistir en una lectura autónoma de la documentación o en un aprendizaje guiado por otra(s) persona(s).
- RNF.6: Los usuarios técnicos no informáticos deberán ser capaces de usar el sistema de forma auxiliar tras un máximo de seis horas de aprendizaje o entrenamiento.
  - RNF.6.1: El uso auxiliar del sistema comprende la siguiente funcionalidad:
    - RNF.6.1.1: Abrir o acceder a la interfaz del sistema.
    - RNF.6.1.2: Ver un listado de los problemas actuales, o la ausencia de estos.
    - RNF.6.1.3: Acceder a un listado de los sensores actualmente conectados.
    - RNF.6.1.4: Ver el estado actual de cada uno de esos sensores.
    - RNF.6.1.5: Acceder a los detalles de conexión para cada uno de esos sensores.
    - RNF.6.1.6: Acceder a la configuración del sistema.
  - RNF.6.2: Dicho aprendizaje o entrenamiento podrá consistir en una lectura autónoma de la documentación o en un aprendizaje guiado por otra(s) persona(s).
- RNF.7: Los requisitos RNF.4, RNF.5 y RNF.6 no impiden que una misma persona pueda ejercer diferentes roles de usuario según su capacidad o las necesidades del momento.
- RNF.8: Desde que un usuario abra o acceda a la interfaz hasta que esta se muestre transcurrirá un máximo de 10 segundos.
  - RNF.8.1: El hecho de mostrar la interfaz de usuario no implica mostrar datos o información sobre los sensores, ya que la comunicación de red puede llevar más tiempo en función del estado de la infraestructura.
- RNF.9: Durante el proceso de carga inicial de datos tras acceder al sistema, la interfaz de usuario no debe quedar bloqueada o atascada.
  - RNF.9.1: Esto no impide que algunas opciones o funcionalidades estén temporalmente desactivadas hasta que el sistema esté preparado para funcionar.
- RNF.10: Una vez que la interfaz está estable, los datos de aquellos sensores que se muestren en un momento dado se actualizarán con una frecuencia no superior a un minuto.
  - RNF.10.1: Si en algún momento de la ejecución la interfaz perdiera la conexión o sufriera otro fallo que impida su funcionamiento, se informará al usuario tras un retraso máximo de dos minutos.
  - RNF.10.2: A partir del momento en que se informa al usuario del error, y hasta que el problema sea solucionado, los requisitos sobre rendimiento serán obviados.

# Apéndices

Incluye otros elementos de interés, como diagramas o modelos de datos. Esta sección se considera **más inestable** que el resto del documento, y puede contener información provisional o inexacta.

Por restricciones específicas del cliente, no se permite almacenar datos en la nube, por lo que la configuración del sistema deberá guardarse en archivos locales.

Tanto la interfaz de usuario como los procesos y servidores remotos pueden tener sus propios ficheros de configuración con los datos necesarios. En cualquier caso, se evitará en lo posible duplicar ajustes.

## Posible formato de configuración

```
{
  Sistema:
  {
    TiempoEspera: 30
    NumIntentos: 3
  }
  Sensores:
  {
    IdConectados: "Sensor1;Sensor2"
  }
  Sensor1:
  {
    DireccionIP: "156.35.0.1"
    Puerto: 5000
    TipoSensor: "TipoA"
  }
  Sensor2:
  {
    DireccionIP: "156.35.0.2"
    Puerto: 5000
    TipoSensor: "TipoB"
  }
}
```

Categoría **Sistema**: Ajustes globales y de la interfaz de usuario.

Categoría **Sensores**: Datos generales sobre los sensores conectados.

Categoría **IdSensor**: Datos necesarios para conectarse y administrar cada uno de los sensores conectados al sistema.

# Detalles sobre los procesos modulares

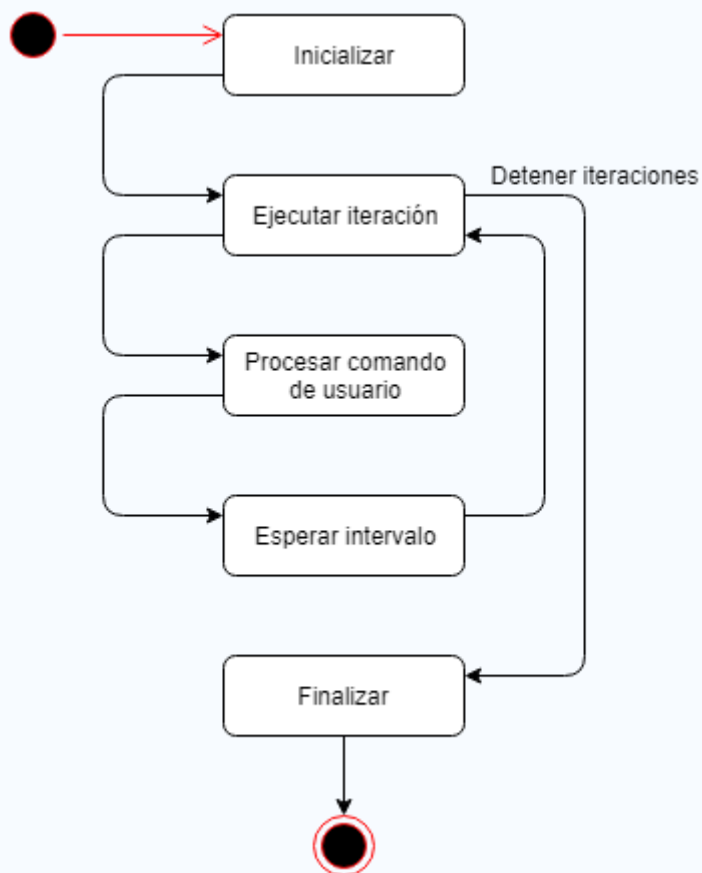
Cada proceso modular está diseñado para **comunicarse con un solo tipo de sensor**, y en principio **no se ejecutará directamente**, sino que el esqueleto anfitrión será el encargado de gestionar su ciclo de vida.

Además, dicho anfitrión no depende en tiempo de compilación de ningún proceso concreto, sino que puede cargar e instanciar en tiempo de ejecución ensamblados DLL que contengan cualquier proceso, siempre que este **cumpla una interfaz común**.

Es factible (aunque aún no está decidido) permitir que **un proceso se comunique con otro** dentro del mismo anfitrión, de forma que un sensor con la suficiente complejidad pudiera necesitar más de un proceso para funcionar. No obstante, lo ideal es que cada proceso sea independiente.

La mayoría de procesos para conseguir datos de sensores estarán diseñados para **ejecutarse de forma continua** (una vez por minuto, por ejemplo), pero otros procesos (de mantenimiento, o para cambiar ajustes durante el desarrollo) pueden tener **una sola iteración** de trabajo y ejecutarse puntualmente.

A grandes rasgos, su ciclo de vida es el siguiente:



- **Inicializar:** Se ejecuta una vez cuando el anfitrión carga el proceso en memoria. En este punto se leerían archivos de configuración o similar.
- **Ejecutar iteración:** Se ejecuta una vez por iteración, y marca el momento en el que el proceso hace un ciclo de trabajo. Por ejemplo, leer un dato nuevo del sensor.
- **Procesar comando de usuario:** El proceso puede recibir mensajes del usuario o de otros procesos. En este paso el anfitrión se los entrega, para que pueda alterar su comportamiento o configuración. Se ejecuta una vez por iteración, pero solo si hay mensajes pendientes.
- **Esperar intervalo:** Después de cada iteración, el proceso espera el intervalo indicado antes de volver al paso 'Ejecutar iteración'
- **Finalizar:** Se ejecuta una vez cuando el anfitrión se dispone a descargar el proceso de memoria. En este punto se liberarían recursos o similar.

Un proceso puede configurar su propio intervalo al inicializarse, o en cualquier momento de su ciclo de vida, pero no se garantiza que el anfitrión lo respete: si el intervalo es muy corto (por ejemplo, menos de un segundo), podría establecer un intervalo mínimo por defecto; si es muy largo (más de una hora), podría establecer un intervalo máximo o incluso convertirlo en un proceso de una sola iteración.

Además, los procesos pueden solicitar su finalización en cualquier momento de su ciclo de vida, pero el anfitrión podría decidir finalizarlos por su cuenta si detectara las circunstancias para ello. En cualquier caso, si un proceso decide finalizar, el anfitrión no puede (o no debería) mantenerlo activo unilateralmente.

La documentación sobre procesos modulares se irá extendiendo a medida que el sistema se desarrolla. En este momento representa lo que debería hacer, y no lo que necesariamente ya hace.

Para más detalles sobre el progreso y funcionamiento del esqueleto anfitrión y los procesos modulares, se puede consultar el repositorio en GitHub donde está almacenado:

<https://github.com/carlubian/Kaomi>