

Nota: La portada aparece en la versión entregable del documento. A efectos de permanencia pública, y por motivos de privacidad, ha sido eliminada de esta versión.

## Agradecimientos

---

El contenido de esta página aparece en la versión entregable del documento. A efectos de permanencia pública, y por motivos de privacidad, ha sido eliminada de esta versión.



# Resumen

---

Este proyecto busca desarrollar una aplicación que permita al usuario ver el estado de un conjunto de sensores distribuidos en sistemas externos. Dicho conjunto será configurable por el usuario desde dentro de la propia aplicación, para permitirle añadir, filtrar y eliminar sensores según sus necesidades.

Está destinado a su uso interno por parte de la empresa cliente, y de acuerdo con sus requisitos, está implementado como una Aplicación Universal de Windows escrita en C#.

La arquitectura del sistema se divide en dos partes:

- La interfaz de usuario, que actúa como cliente y se ejecuta sobre el sistema local.
- Los módulos de sensores, que actúan como servidores desplegados en los sistemas remotos donde estén conectados los sensores.

Durante la ejecución, la interfaz pedirá datos a los módulos remotos de forma periódica para mostrar al usuario la situación actual de todos los sensores conectados.

El principal beneficio de esta separación es que, al usar un sistema de comunicación común, es posible añadir nuevos tipos de sensor sin necesidad de modificar la interfaz de usuario, que se limita a recibir y mostrar datos independientemente del hardware concreto del que procedan.

# Palabras Clave

---

Sensores, Monitorización, Cliente-Servidor, .NET, HTTP, UWP.

# Abstract

---

This project seeks to develop an application that will allow the user to see the status of a group of sensors distributed across external systems. Said group will be configurable by the user from within the application itself, as to allow them to add, filter and remove sensors according to their needs.

It is intended for internal use by the client company, and per their requirements, it's implemented as a Universal Windows App written in C#.

The system architecture is divided into two parts:

- The user interface, acting as a client and running on the local system.
- The sensor modules, acting as servers deployed on the remote systems the sensors are connected to.

During execution, the interface will request data from the remote modules periodically to show the user the current status of all connected sensors.

The main benefit of this separation is that, by using a common communications system, it is possible to add new types of sensors without needing to modify the user interface, that will simply receive and display data regardless of the specific underlying hardware.

## Keywords

---

Sensors, Monitorization, Client-Server, .NET, HTTP, UWP.

# Índice General

<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>13</b>
1.1 JUSTIFICACIÓN DEL PROYECTO .....	13
1.2 OBJETIVOS DEL PROYECTO.....	15
1.3 ESTUDIO DE HERRAMIENTAS EXISTENTES.....	16
1.3.1 <i>SmartSense de Digi International</i> .....	16
1.3.2 <i>Dashboard de Airthings</i> .....	17
1.3.3 <i>Senaps de Data61</i> .....	18
1.3.4 <i>Conclusión</i> .....	19
1.4 EVALUACIÓN DE ALTERNATIVAS DE DESARROLLO .....	19
1.4.1 <i>Interfaz web</i> .....	19
1.4.2 <i>Aplicación de consola</i> .....	20
1.4.4 <i>Aplicación UWP sobre C#</i> .....	20
1.4.5 <i>Conclusión</i> .....	21
<b>CAPÍTULO 2. ASPECTOS TEÓRICOS.....</b>	<b>22</b>
2.1 DATOS COMUNES .....	22
2.2 SENSOR GOCATOR DE LMI TECHNOLOGIES .....	22
2.3 CÁMARA ESPECTRAL sCMOS-50-V10E DE SPECIM SPECTRAL IMAGING.....	24
2.4 SENSOR DE PARTÍCULAS ISOAIR DE SPECTRIS.....	24
<b>CAPÍTULO 3. PLANIFICACIÓN DEL PROYECTO Y PRESUPUESTOS.....</b>	<b>27</b>
3.1 PLANIFICACIÓN .....	27
3.1.1 <i>Diagramas de Gantt</i> .....	28
3.2 PRESUPUESTO.....	29
3.2.1 <i>Presupuesto de costes</i> .....	29
3.2.2 <i>Presupuesto de cliente</i> .....	30
<b>CAPÍTULO 4. ANÁLISIS .....</b>	<b>31</b>
4.1 DEFINICIÓN DEL SISTEMA.....	31
4.1.1 <i>Determinación del Alcance del Sistema</i> .....	31
4.2 REQUISITOS DEL SISTEMA.....	32
4.2.1 <i>Obtención de los Requisitos del Sistema</i> .....	32
4.2.2 <i>Identificación de Actores del Sistema</i> .....	44
4.2.3 <i>Especificación de Casos de Uso</i> .....	45
4.3 IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS.....	47
4.3.1 <i>Descripción de los Subsistemas</i> .....	47
4.3.2 <i>Descripción de los Interfaces entre Subsistemas</i> .....	47
4.4 DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS.....	48
4.4.1 <i>Diagrama de Clases</i> .....	48
4.4.2 <i>Descripción de las Clases</i> .....	49
4.5 ANÁLISIS DE CASOS DE USO Y ESCENARIOS .....	57

4.5.1	<i>Gestionar sensores conectados</i>	57
4.5.2	<i>Ver estado de un sensor</i>	61
4.5.3	<i>Modificar la configuración del sistema</i>	61
4.6	ANÁLISIS DE INTERFACES DE USUARIO	63
4.6.1	<i>Descripción de la Interfaz</i>	63
4.6.2	<i>Descripción del Comportamiento de la Interfaz</i>	68
4.6.3	<i>Diagrama de Navegabilidad</i>	69
4.7	ESPECIFICACIÓN DEL PLAN DE PRUEBAS	70
4.7.1	<i>Pruebas Unitarias</i>	70
4.7.2	<i>Pruebas de Integración y del Sistema</i>	74
4.7.3	<i>Pruebas de Usabilidad</i>	76
4.7.4	<i>Pruebas de Rendimiento</i>	79
<b>CAPÍTULO 5. DISEÑO DEL SISTEMA</b>		<b>80</b>
5.1	ARQUITECTURA DEL SISTEMA	80
5.1.1	<i>Diagramas de Paquetes</i>	80
5.1.2	<i>Diagramas de Despliegue</i>	81
5.2	DISEÑO DE CLASES	82
5.2.1	<i>TFG.UWP</i>	82
5.2.2	<i>TFG.Core</i>	84
5.2.3	<i>Kaomi.Core</i>	85
5.2.4	<i>Kaomi.WebAPI</i>	86
5.2.5	<i>ConfigAdapter</i>	87
5.2.6	<i>ConfigAdapter.Xml</i>	88
5.3	DISEÑO DE LA INTERFAZ	89
<b>CAPÍTULO 6. IMPLEMENTACIÓN DEL SISTEMA</b>		<b>93</b>
6.1	ESTÁNDARES Y NORMAS SEGUIDOS	93
6.2	LENGUAJES DE PROGRAMACIÓN	94
6.3	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO	95
6.3.1	<i>C#</i>	95
6.3.2	<i>Visual Studio</i>	95
6.3.3	<i>Visual Studio Code</i>	95
6.3.4	<i>Framework .NET</i>	95
6.3.5	<i>.NET Core</i>	95
6.3.6	<i>ASP.NET Core</i>	95
6.3.7	<i>Plataforma UWP</i>	96
6.3.8	<i>Azure DevOps</i>	96
6.3.9	<i>NuGet</i>	96
6.3.10	<i>Mapas de Bing</i>	96
6.3.11	<i>Peticiones REST</i>	96
6.4	CREACIÓN DEL SISTEMA	97
6.4.1	<i>Problemas Encontrados</i>	97
<b>CAPÍTULO 7. DESARROLLO DE LAS PRUEBAS</b>		<b>99</b>
7.1	PRUEBAS UNITARIAS	99



7.2	PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA.....	105
7.3	PRUEBAS DE USABILIDAD .....	109
7.3.1	<i>Usuario de pruebas 1.....</i>	<i>109</i>
7.3.2	<i>Usuario de pruebas 2.....</i>	<i>110</i>
7.3.3	<i>Usuario de pruebas 3.....</i>	<i>111</i>
7.3.4	<i>Observaciones.....</i>	<i>113</i>
7.4	PRUEBAS DE RENDIMIENTO.....	114
7.4.1	<i>Conclusiones .....</i>	<i>114</i>
<b>CAPÍTULO 8. MANUALES DEL SISTEMA .....</b>		<b>115</b>
8.1	MANUAL DE INSTALACIÓN.....	115
8.1.1	<i>Instalación del servidor remoto.....</i>	<i>115</i>
8.1.2	<i>Instalación de la aplicación local.....</i>	<i>116</i>
8.2	MANUAL DE USUARIO .....	118
8.2.1	<i>Primer uso .....</i>	<i>118</i>
8.2.2	<i>Funciones principales.....</i>	<i>121</i>
8.2.3	<i>Acceso Rápido .....</i>	<i>123</i>
8.3	MANUAL DEL PROGRAMADOR .....	125
8.3.1	<i>Crear un proceso modular nuevo.....</i>	<i>125</i>
<b>CAPÍTULO 9. CONCLUSIONES Y OTRAS AMPLIACIONES.....</b>		<b>129</b>
9.1	CONCLUSIONES.....	129
9.2	AMPLIACIONES.....	129
9.2.1	<i>Mejorar las capacidades del mapa.....</i>	<i>129</i>
9.2.2	<i>Registrar notificaciones.....</i>	<i>129</i>
9.2.3	<i>Internacionalizar la aplicación.....</i>	<i>129</i>
<b>CAPÍTULO 10. REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>131</b>
10.1	REFERENCIAS EN INTERNET .....	131



# Índice de Figuras

Figura 1.1. Captura de la aplicación SmartSense .....	16
Figura 1.2. Captura de la aplicación Dashboard .....	17
Figura 1.3. Captura de la aplicación Senaps.....	18
Figura 2.1. Funcionamiento de un sensor láser Gocator .....	23
Figura 3.1. Diagrama de Gantt inicial.....	28
Figura 3.2. Diagrama de Gantt final .....	28
Figura 4.1. Diagrama de casos de uso.....	45
Figura 4.2. Diagrama de clases preliminar .....	48
Figura 4.3. Esquema de la interfaz de usuario .....	63
Figura 4.4. Esquema de la pantalla principal .....	64
Figura 4.5. Esquema de la pantalla de listado .....	65
Figura 4.6. Esquema de la pantalla de configuración.....	65
Figura 4.7. Esquema de la pantalla de conexión de un sensor .....	66
Figura 4.8. Esquema de la pantalla de detalles de un sensor .....	66
Figura 4.9. Esquema de la pantalla de modificación de un sensor .....	67
Figura 4.10. Esquema de la interfaz del Acceso Rápido .....	67
Figura 4.11. Diagrama de navegabilidad.....	69
Figura 5.1. Diagrama de paquetes del proyecto.....	80
Figura 5.2. Diagrama de despliegue del sistema.....	81
Figura 5.3. Diagrama de clases principal del paquete TFG.UWP .....	82
Figura 5.4. Diagrama de clases de diálogos del paquete TFG.UWP .....	83
Figura 5.5. Diagrama de clases principal del paquete TFG.Core .....	84
Figura 5.6. Diagrama de clases principal del paquete Kaomi.Core.....	85
Figura 5.7. Diagrama de clases principal del paquete Kaomi.WebAPI.....	86
Figura 5.8. Diagrama de clases principal del paquete ConfigAdapter .....	87
Figura 5.9. Diagrama de clases principal del paquete ConfigAdapter.Xml.....	88
Figura 5.10: Pantalla inicial de la aplicación. ....	89
Figura 5.11: Pantalla de listado de sensores.....	89
Figura 5.12: Diálogo para filtrar el listado.....	90
Figura 5.13: Pantalla de detalles de un sensor. ....	90
Figura 5.14: Pantalla de modificación de un sensor.....	91
Figura 5.15: Pantalla de conexión de un sensor nuevo.....	91
Figura 5.16: Pantalla de ajustes del sistema. ....	92
Figura 5.17: Panel de Acceso Rápido .....	92
Figura 7.1: Pantalla de detalles de un sensor.....	113
Figura 8.1. Experiencia de Nuevo Usuario .....	118
Figura 8.2. Vista inicial de la página principal.....	119
Figura 8.3. Paso 1 de la conexión de nuevos sensores .....	119
Figura 8.4. Paso 2 de la conexión de nuevos sensores .....	120
Figura 8.5. Página principal con un sensor conectado .....	120
Figura 8.6. Listado inicial de sensores .....	121

Figura 8.7. Diálogo de filtros y ordenación.....	121
Figura 8.8. Detalles de un sensor desconectado .....	122
Figura 8.9. Detalles de un sensor conectado.....	122
Figura 8.10. Modificación de los datos de un sensor .....	123
Figura 8.11. Panel de Acceso Rápido .....	123

# Capítulo 1. Introducción

## 1.1 Justificación del Proyecto

El cliente de este proyecto tiene, entre su infraestructura, varios dispositivos sensores instalados en ordenadores a lo largo del mundo. Estos sensores están destinados a uso industrial, lo que implica que utilizan sistemas de comunicaciones menos ‘habituales’ o ‘cómodos’ que los sensores destinados a uso doméstico. En concreto:

- La mayoría de sensores domésticos suelen incluir una aplicación que permite ver los datos recogidos por varios sensores de la misma marca. En el caso de sensores de uso industrial, esta posibilidad no suele ofrecerse. Es responsabilidad de los usuarios finales conectarse y obtener datos de cada sensor.
- Los sensores domésticos normalmente utilizan sistemas comunes y fáciles de configurar para comunicarse, como Wifi o Bluetooth. En cambio, los sensores industriales prefieren conexiones cableadas (ya sea sobre Ethernet o fibra óptica) para transmitir datos mediante protocolos como MQTT, OPC o Sockets.

Se presuponen las siguientes condiciones iniciales:

- El cliente tiene sensores de diferentes tipos, que no utilizan un sistema común para transmitir datos.
- Todos los ordenadores con sensores instalados están conectados a la red interna del cliente.
- El cliente desea controlar una cantidad máxima de entre 15 y 30 sensores mediante este sistema.

Antes del desarrollo de este proyecto, cada uno de los sensores utilizados por el cliente dispone de una interfaz individual, por lo que comprobar el estado de todos ellos es un proceso largo e incómodo.

Además, estas interfaces tienen características diferentes (al estar desarrolladas por las empresas responsables de cada sensor) y su experiencia de usuario no está relacionada, por lo que es necesario conocer las particularidades de cada una de ellas para poder usarlas de forma eficaz.

El proyecto consistirá en la creación de una aplicación unificada para mostrar el estado de estos sensores en una sola plataforma. De esta manera, se permite apreciar de un vistazo la situación global, en vez de tener que usar los sistemas propios de cada sensor.

Las características clave deseadas son las siguientes:

- Poder añadir y eliminar sensores (potencialmente de diferentes tipos) sin tener que modificar el código de la aplicación.
- Poder filtrar y ordenar los sensores mostrados en función de criterios significativos.
- Poder ver el estado operacional de los sensores (conectado, desconectado, versión del software), sin incluir los datos recogidos por los mismos (que varían en función del propósito de cada sensor).

## 1.2 Objetivos del Proyecto

Los objetivos principales que se busca cumplir con este proyecto son los siguientes:

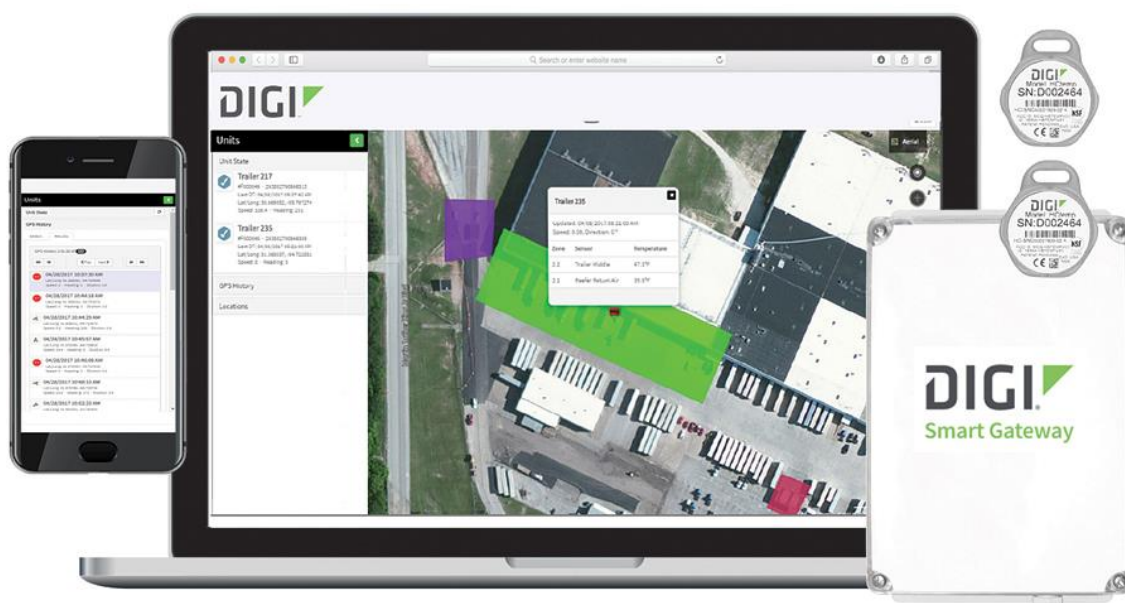
- Permitir al usuario gestionar la conexión con un conjunto de sensores de diferentes tipos desde una sola aplicación.
- Permitir al usuario comprobar el estado de los sensores conectados.
- Notificar al usuario cuando alguno de los sensores conectados cambie de estado debido a un error o una desconexión.

## 1.3 Estudio de Herramientas existentes

A continuación, se evaluarán algunas alternativas y posibilidades para el desarrollo del sistema, indicando sus ventajas o inconvenientes con respecto a la opción elegida finalmente.

### 1.3.1 SmartSense de Digi International

Esta plataforma permite a las empresas gestionar sensores y recibir alertas mediante un sistema basado en la nube.



*Figura 1.1. Captura de la aplicación SmartSense*

Las ventajas de este sistema son:

- Compatibilidad: Promete ser compatible con todos los sensores necesarios.

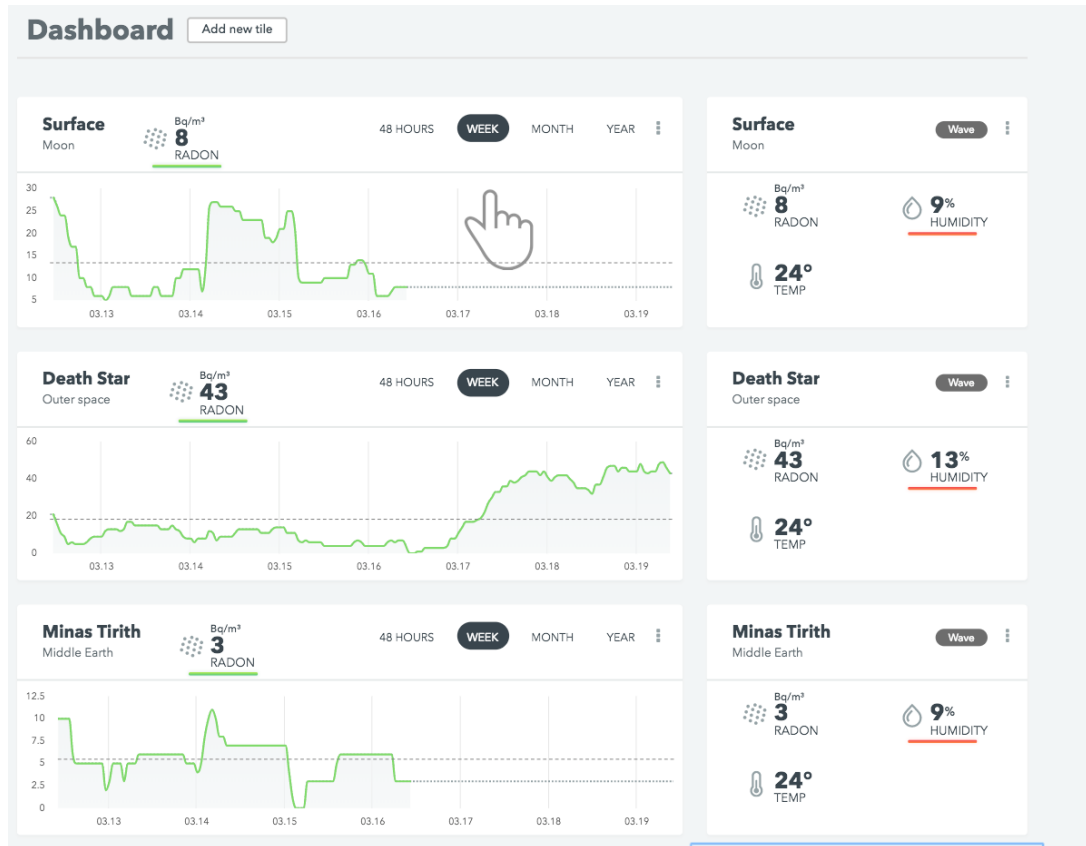
Los inconvenientes de este sistema son:

- Infraestructura: Al estar creado por un tercero, se depende de él para que dé servicio.
- Comunicaciones: Este sistema necesita almacenar los datos en sus propios servidores. Las normas de seguridad del cliente impiden utilizar proveedores externos para almacenar datos.



## 1.3.2 Dashboard de Airthings

Esta aplicación ofrece la posibilidad de conectar varios sensores y recibir sus datos en una interfaz web.



*Figura 1.2. Captura de la aplicación Dashboard*

Las ventajas de este sistema son:

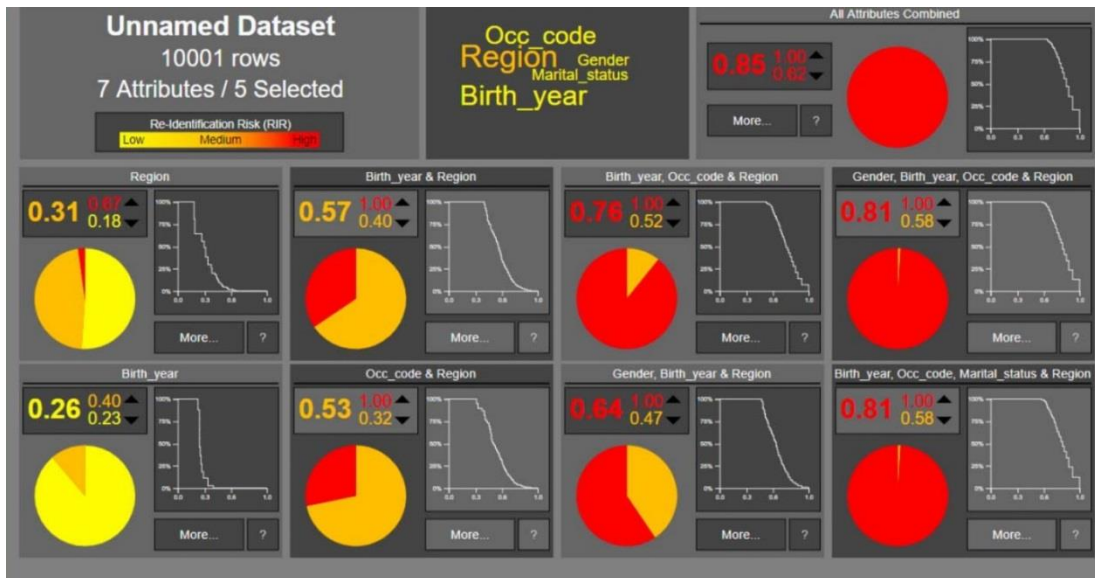
- Funcionalidad: Permite ver datos históricos, y exportarlos para un análisis posterior.

Los inconvenientes de este sistema son:

- Compatibilidad: Parece estar enfocado a sensores de uso doméstico, en vez de los sensores industriales usados por el cliente.
- Comunicaciones: Los sensores conectados deben ser compatibles por sí mismos con los protocolos usados por la aplicación. En caso contrario, la conexión no parece a priori posible.

### 1.3.3 Senaps de Data61

Este sistema está enfocado al procesamiento de Big Data desde una nube de sensores, para luego mostrar la información agregada en aplicaciones personalizadas.



*Figura 1.3. Captura de la aplicación Senaps*

Las ventajas de este sistema son:

- Funcionalidad: Los servicios usados para procesar datos son fácilmente escalables y ofrecen un buen rendimiento.
- Flexibilidad: El flujo de datos es configurable, para modificar los criterios de procesamiento o la forma final de mostrarlos.

Los inconvenientes de este sistema son:

- Comunicaciones: Este sistema almacena y procesa los datos en sus propios servidores. Las normas de seguridad del cliente impiden usar proveedores externos para este fin.
- Infraestructura: Este sistema está pensado para tratar cantidades de sensores mucho mayores de las utilizadas por el cliente (del orden de miles), por lo que se acabaría tratando con un exceso de costes e infraestructuras innecesarias.

## 1.3.4 Conclusión

En esta tabla se puede comprobar de forma rápida las ventajas e inconvenientes de cada opción estudiada en función de cada criterio considerado:

	C1	C2	C3	C4	C5
SmartSense	✓	✗		✗	
AirThings	✗	✗		✓	
Senaps		✗	✓	✗	✓

- C1: Compatibilidad
- C2: Comunicaciones
- C3: Funcionalidad
- C4: Infraestructura
- C5: Flexibilidad

Como se puede ver, ninguna de las alternativas analizadas cumple todos los requisitos. Por tanto, es necesario desarrollar un nuevo sistema.

## 1.4 Evaluación de alternativas de desarrollo

Pese a que el cliente impone como entorno de desarrollo la plataforma .NET, se han analizado otras alternativas por si se encontrara alguna con mayores ventajas.

### 1.4.1 Interfaz web

Una posibilidad bastante popular hoy en día sería implementar el sistema como una aplicación web, o como alternativa similar, crear una interfaz web sobre un servicio o aplicación núcleo sin interfaz.

Las ventajas de este sistema son:

- Compatibilidad: En principio las aplicaciones web son compatibles con la mayoría de sistemas operativos y navegadores, facilitando su utilización a los usuarios.
- Infraestructura: La mayoría de aplicaciones web tienen un menor uso de recursos en comparación con aplicaciones tradicionales.

Los inconvenientes de este sistema son:

- Funcionalidad: Por el hecho de ser compatibles con todas las plataformas, no es sencillo utilizar mecanismos nativos como notificaciones u otras funciones específicas de una plataforma concreta.

## 1.4.2 Aplicación de consola

Las aplicaciones de consola son una alternativa sencilla cuando se trata de operaciones repetitivas o de bajo nivel.

Las ventajas de este sistema son:

- Flexibilidad: Al trabajar únicamente con texto, es relativamente sencillo mostrar unos datos u otros, simplemente cambiando el texto que se muestra.
- Compatibilidad: Prácticamente todos los sistemas operativos en el mercado tienen un sistema de consola disponible.

Las desventajas de este sistema son:

- Funcionalidad: Las aplicaciones de consola no están diseñadas para crear experiencias de usuario visualmente atractivas, y pueden volverse incómodas si hay que mostrar una gran cantidad de datos.
- Necesidades del cliente: Al cliente de este proyecto no le ilusiona particularmente utilizar aplicaciones de consola, y prefiere una interfaz de usuario más moderna.

## 1.4.4 Aplicación UWP sobre C#

Las ventajas de este sistema son:

- Infraestructura: Debido a su popularidad, la mayoría de sensores tienen librerías disponibles. En caso de necesidad, es posible integrar también código diseñado para C++ a través del framework .NET
- Comunicaciones: Es posible utilizar varios sistemas de almacenamiento local, como archivos de configuración o bases de datos embebidas, que mantienen los datos bajo control.
- Funcionalidad: Este entorno tiene un sistema unificado y estandarizado para diseñar interfaces de usuario, soportado directamente por Microsoft.
- Flexibilidad: Al ser relativamente reciente, este entorno incluye muchas facilidades a la hora de desarrollar y mantener aplicaciones. Además, las herramientas de desarrollo son de gran calidad, lo que mejora la eficiencia en este aspecto.
- Necesidades del cliente: El cliente está acostumbrado a trabajar con este entorno, y dispone de suficiente personal cualificado para mantenerlo y utilizarlo.
- Compatibilidad: Además de tener compatibilidad nativa con Windows, las nuevas versiones son multiplataforma.

## 1.4.5 Conclusión

En esta tabla se puede comprobar de forma rápida las ventajas e inconvenientes de cada opción estudiada en función de cada criterio considerado:

	C1	C2	C3	C4	C5
Interfaz web	✓		✗	✓	
Consola	✓		✗		✓
Aplicación UWP	✓	✓	✓	✓	✓

- C1: Compatibilidad
- C2: Comunicaciones
- C3: Funcionalidad
- C4: Infraestructura
- C5: Flexibilidad

Por lo tanto, el sistema elegido es una aplicación UWP sobre C#.

## Capítulo 2. Aspectos Teóricos

En esta sección se explicarán a grandes rasgos algunos elementos del dominio utilizados durante el desarrollo del proyecto, incluyendo sus características y el propósito de su uso.

### 2.1 Datos comunes

Con independencia del tipo de sensor, hay un conjunto de datos comunes a todos ellos. Por supuesto, será posible mostrar datos adicionales en función del tipo de sensor concreto:

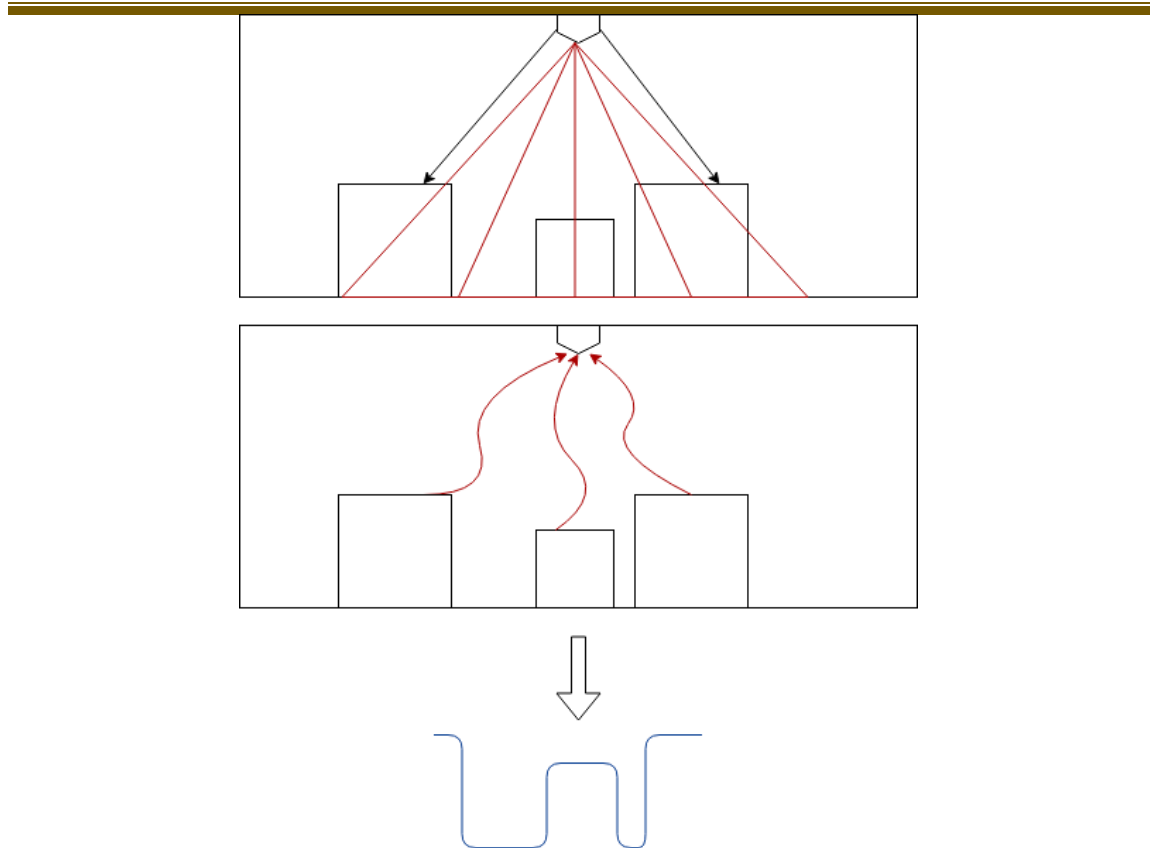
- **Conectado:** Indica si el sensor está 'activo', es decir, tomando lecturas o haciendo mediciones. El hecho de que un sensor esté desconectado no implica que esté apagado, sino que podría estar en reposo o a la espera.
- **Compatible:** Indica si el sensor está utilizando un firmware válido y actualizado. Un valor falso en esta propiedad no implica errores en este sistema, sino un problema con el propio dispositivo.
- **Estado:** Contiene un valor específico de cada tipo de sensor que indica su estado actual. Normalmente será coherente con los dos valores anteriores, y podría ampliar la información de los mismos.

### 2.2 Sensor Gocator de LMI Technologies

Este tipo de dispositivo consiste en un emisor láser y una cámara de recepción montados sobre un chasis metálico. Su principal propósito es medir perfiles o profundidades.

Funciona de forma similar a un radar: El emisor láser lanza una línea de ondas que abarca un máximo de 1m de ancho. Estas ondas láser rebotan con los obstáculos que se encuentran y vuelven hacia el sensor, siendo captados por la cámara.

El resultado es un perfil que representa la forma física de los objetos dentro del campo visual del sensor:



*Figura 2.1. Funcionamiento de un sensor láser Gocator*

Este tipo de sensor transmite los siguientes datos, además de los datos comunes mencionados anteriormente:

- **Modelo:** Una cadena de texto con el nombre del modelo concreto, que varía en función de la serie. Las diferencias entre un modelo y otro suelen ser aspectos técnicos como la resolución o frecuencia máxima, pero no afectan a este sistema.
- **Protocolo:** Una serie de números que indican la versión del protocolo utilizado para transmitir datos. Dicho protocolo queda abstraído por las librerías del fabricante, y no provoca problemas de compatibilidad.
- **Firmware:** Una serie de números que indican la versión del firmware instalado en el sensor.
- **Modo de escaneo:** Indica el modo de funcionamiento del sensor. La mayoría de modelos pueden tanto leer perfiles lineales como superficies rectangulares.
- **Alineación:** Indica si el emisor láser y la cámara están alineados. Si el sensor ha sido desplazado físicamente, puede ser necesario alinearlo de nuevo para que las lecturas sean consistentes.

- **Valor de encoder:** Un número que indica los milisegundos transcurridos desde el comienzo de las lecturas. Se puede utilizar como magnitud de tiempo a la hora de interpretar los datos.
- **Perfil de trabajo:** Una cadena de texto que hace referencia a la configuración actual del sensor. Estos sensores son capaces de guardar varias configuraciones, y cambiar de una a otra.

## 2.3 Cámara espectral sCMOS-50-V10E de Specim Spectral Imaging

Este dispositivo consiste en una cámara capaz de captar datos espectrales además de imágenes naturales. Estos datos pueden utilizarse para calcular, entre otras posibilidades, los siguientes resultados:

- Imágenes tridimensionales, calculando la distancia de cada píxel a la cámara.
- Intensidades o tonalidades de color, midiendo las longitudes de onda que son absorbidas o reflejadas.
- Diferenciar entre tipos de material, comparando imágenes infrarrojas.

Además de los datos comunes, es posible obtener los siguientes datos de este tipo de sensores:

- **Rango espectral:** Un número que representa una magnitud en nanómetros, normalmente entre 400 y 1000. Indica la separación entre puntos de lectura.
- **Tiempo de exposición:** Un número que indica el tiempo de exposición de cada lectura.
- **Temperatura de funcionamiento:** Algunos modelos incluyen un sensor de temperatura que permite monitorizar las condiciones de trabajo. Este tipo de dispositivos puede funcionar entre -20 y +50 grados centígrados.

## 2.4 Sensor de partículas IsoAir de Spectris

Este tipo de sensores son capaces de medir concentraciones de partículas atmosféricas de diversos elementos. Incluyen su propio sistema de ventilación, de forma que no es necesario instalar aspiradores de aire en la ubicación del mismo.

Además, es posible conectar hasta tres entradas de gas en un mismo sensor, para aumentar la cobertura y evitar tanto falsos positivos como falsos negativos.

Estos sensores son capaces de transmitir los siguientes datos, además de los datos comunes:



- **Valor de histéresis:** Un número que indica la concentración mínima de partículas detectable. Esto se usa para filtrar ruido de fondo o mitigar lecturas incorrectas.
- **Rango de detección:** Indica los márgenes de concentración de partículas que son detectados, que dependen del modelo concreto de sensor. Si la concentración es menor que el valor mínimo, no se detecta; Si es mayor que el valor máximo, el sensor puede saturarse y enviar lecturas inconsistentes.



## Capítulo 3. Planificación del Proyecto y Presupuestos

### 3.1 Planificación

La planificación inicial del proyecto, basada en estimaciones, fue la siguiente:

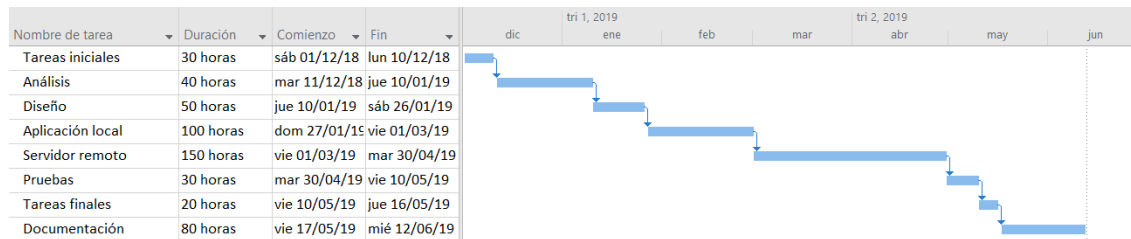
	Horas estimadas	Fecha de inicio	Fecha de finalización
Tareas iniciales	30	1-DIC-2018	10-DIC-2018
Análisis	40	11-DIC-2018	10-ENE-2019
Diseño	50	10-ENE-2019	26-ENE-2019
Aplicación local	100	27-ENE-2019	1-MAR-2019
Servidor remoto	150	1-MAR-2019	30-ABR-2019
Pruebas	30	30-ABR-2019	10-MAY-2019
Tareas finales	20	10-MAY-2019	16-MAY-2019
Documentación	80	17-MAY-2019	12-JUN-2019

Sin embargo, durante el proceso se hizo patente que las estimaciones se quedarían cortas, por lo que fue necesario modificar la planificación, que quedó de la siguiente manera:

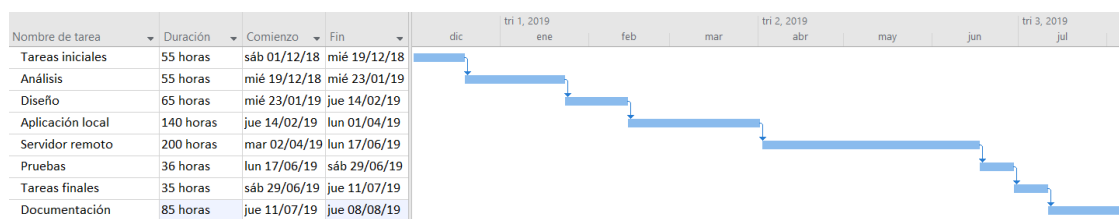
	Horas reales	Fecha de inicio	Fecha de finalización
Tareas iniciales	55	1-DIC-2018	19-DIC-2018
Análisis	55	19-DIC-2018	23-ENE-2019
Diseño	65	23-ENE-2019	14-FEB-2019
Aplicación local	140	14-FEB-2018	1-ABR-2018
Servidor remoto	200	2-ABR-2019	17-JUN-2019
Pruebas	36	17-JUN-2019	29-JUN-2019
Tareas finales	35	29-JUN-2019	11-JUL-2019
Documentación	85	11-JUL-2019	8-AGO-2019

### 3.1.1 Diagramas de Gantt

Estos diagramas permiten comprobar de forma visual la evolución de cada tarea del proyecto, así como las diferencias entre la planificación inicial y la final:



*Figura 3.1. Diagrama de Gantt inicial*



*Figura 3.2. Diagrama de Gantt final*

## 3.2 Presupuesto

### 3.2.1 Presupuesto de costes

En esta sección se incluye el presupuesto de costes sobre la planificación inicial, que representa los gastos reales derivados de desarrollar el proyecto.

Las horas de implementación se han obtenido mediante una estimación por Puntos Función:

	Aplicación local	Servidor remoto
PF no ajustados	46	54
PF ajustados	40	50
Horas por PF	2,5	3
Horas de desarrollo	100	150

El coste de la hora de trabajo se ha establecido de acuerdo al coste estándar de cada actividad.

Concepto	Cantidad	Precio unidad	Precio total
<b>Costes de personal</b>			
Análisis (horas)	51	60	3.060,00 €
Diseño (horas)	54	60	3.240,00 €
Pruebas (horas)	36	25	900,00 €
Documentación (horas)	78	25	1.950,00 €
<b>Costes de implementación</b>			
Aplicación local (horas)	120	30	3.600,00 €
Servidor remoto (horas)	188	30	5.640,00 €
<b>Software</b>			
Microsoft Project 2016	1	850	850,00 €
Visual Studio Professional 2019 (mes)	10	45	450,00 €
Microsoft Office 365 (mes)	10	10	100,00 €
<b>Hardware</b>			
Ordenador de desarrollo	1	1200	1.200,00 €
Sensor de pruebas (horas)	8	10	80,00 €
<b>Otros costes</b>			
Suministros de oficina	1	50	50,00 €
Facturas y gastos variados (mes)	10	200	2.000,00 €
<b>Coste total</b>			23.120,00 €

## 3.2.2 Presupuesto de cliente

En esta sección se incluye el presupuesto de cliente. Su principal particularidad es que solo muestra aquellos elementos útiles para el cliente. El coste de los demás elementos (además de un margen de beneficio del 20%) se reparte entre los restantes.

Concepto	Cantidad	Precio unidad
<b>Desarrollo del sistema</b>	1	
Análisis		4.616,46 €
Diseño		4.888,01 €
Implementación		13.939,89 €
Pruebas		1.357,78 €
Documentación		2.941,86 €
<b>Subtotal</b>		27.744,00 €
IVA (21%)		5.826,24 €
<b>Total</b>		33.570,24 €

## Capítulo 4. Análisis

Este apartado contendrá toda la especificación de requisitos y toda la documentación del análisis de la aplicación, a partir de la cual se elaborará posteriormente el diseño.

### 4.1 Definición del Sistema

#### 4.1.1 Determinación del Alcance del Sistema

El sistema deberá comunicarse con un número indeterminado y variable de sensores, previamente instalados y en funcionamiento en el momento de la conexión. Estos sensores están 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.

El sistema se adaptará a los tipos de sensor que sean considerados durante la planificación del mismo. No obstante, la arquitectura del sistema estará diseñada de tal forma que sea fácil integrar nuevos tipos de sensor. Para ello, se establecerá un sistema de comunicaciones capaz de transmitir datos arbitrarios desde cualquier sensor.

De esta forma, para integrar un nuevo tipo de sensor, bastará con crear un módulo que implemente el sistema de comunicaciones y transmita los datos con el formato apropiado a la aplicación local, que será capaz de mostrarlos al usuario.

Hay que tener en cuenta que algunos tipos de sensor no considerados durante el desarrollo podrían necesitar transmitir datos en un formato imprevisto, que la aplicación local no sea capaz de mostrar con los medios a su alcance. En este caso, la integración de ese tipo de sensor requerirá modificaciones al sistema no cubiertas en este trabajo.

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

## 4.2 Requisitos del Sistema

### 4.2.1 Obtención de los Requisitos del Sistema

Como fuente inicial de información, se ha partido de la siguiente lista de funcionalidades deseadas por el cliente:

- **Ver el estado global del sistema:** Al abrir la aplicación, el usuario verá una pantalla que le informará del estado de todos los sensores conectados, agrupados por criterios significativos. Se indicarán aquellos que tengan problemas o con los que sea imposible establecer una conexión.
- **Filtrar y ordenar los sensores:** El usuario podrá ver una lista de todos los sensores conectados. Podrá modificar el orden en el que se muestran, además de filtrar un subconjunto de los mismos a través de criterios significativos.
- **Conectar un nuevo sensor al sistema:** Suponiendo que el software necesario esté desplegado en la máquina remota donde se encuentra el sensor, el usuario podrá conectarlo al sistema. Para ello deberá especificar los datos necesarios sobre el sensor.
- **Desconectar un sensor del sistema:** El usuario podrá desconectar alguno de los sensores existentes. Antes de completar la desconexión, se le pedirá confirmación. El hecho de desconectar un sensor no afecta al software desplegado en la máquina remota.
- **Modificar la configuración de un sensor:** El usuario podrá modificar los criterios significativos de alguno de los sensores conectados. Esto afectará a las operaciones de ordenación y filtrado a partir de ese momento.
- **Ver detalles de un sensor:** Desde el listado de sensores, el usuario podrá entrar a una vista detallada del estado de un sensor, incluyendo parámetros técnicos de la conexión y mensajes de estado o error de más bajo nivel.
- **Recibir notificaciones de errores:** Si alguno de los sensores conectados al sistema enviara datos incorrectos o perdiera su conexión, el usuario recibirá una notificación del programa informándole de este evento. Para ello, el programa debe estar ejecutándose.



### 4.2.1.1 Requisitos funcionales

#### 4.2.1.1.1 Aplicación local

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 distribución geográfica de los sensores.

RF.2.1: En dicha distribución, el sistema mostrará un icono para cada país que tenga asociado al menos un sensor.

RF.2.2: El aspecto de estos iconos dependerá del estado de los sensores asociados a cada país. El sistema distinguirá los siguientes casos:

RF.2.2.1: Que al menos uno de los sensores de ese país esté conectado y tenga errores.

RF.2.2.2: Que todos los sensores de ese país estén desconectados.

RF.2.2.3: Que alguno de los sensores de ese país esté desconectado (y que ninguno de los que están conectados tenga errores).

RF.2.2.4: Cualquier situación que no cumpla ninguna de las condiciones anteriores.

RF.3: El sistema permitirá a los usuarios acceder a un listado de todos los sensores conectados.

RF.3.1: En este caso, los sensores estarán ordenados por país sin ningún criterio de filtrado activo.

RF.4: El sistema permitirá a los usuarios acceder a un listado de los sensores asociados a un país determinado.

RF.4.1: En este caso, los sensores estarán ordenados por tipo de sensor y filtrados por país.

RF.5: El sistema permitirá a los usuarios aplicar filtros para ver solo un subconjunto de los listados.

RF.5.1: El sistema mostrará en todo momento el criterio de filtrado y ordenación activo.

RF.5.2: El hecho de aplicar más de un filtro se interpretará como una operación lógica AND. Es decir, solamente los sensores que cumplan todos los criterios activos serán mostrados.

RF.5.3: El sistema ofrecerá los siguientes criterios de filtrado:

RF.5.3.1: País de instalación.

RF.5.3.2: Tipo del sensor.

RF.5.3.3: Lugar de instalación.

RF.5.3.3.1: El lugar de instalación podrá tomar los valores 'Oficina', 'Planta', 'Laboratorio', 'Indefinido'.

RF.5.3.4: Modo de operación.

RF.5.3.4.1: El modo de operación podrá tomar los valores 'Pruebas', 'Producción', 'Indefinido'.

RF.6: El sistema permitirá al usuario modificar o eliminar filtros previamente aplicados, de uno en uno.

RF.7: El sistema permitirá al usuario eliminar todos los filtros aplicados a la vez.

RF.8: El usuario deberá confirmar su selección de filtros para que la vista de datos se actualice.

RF.9: Los usuarios podrán añadir nuevos sensores al sistema.

RF.9.1: Estos sensores deberán tener un servidor remoto compatible con el sistema.

RF.9.2: Para ello, los usuarios deberán indicar los siguientes parámetros de conexión:

RF.9.2.1: Dirección IP.

RF.9.2.2: Puerto.

RF.9.2.3: Tipo de sensor.

RF.9.2.4: País de instalación.

RF.9.2.5: Ciudad de instalación.

RF.9.2.5.1: Este valor será opcional.

RF.9.2.6: Lugar de instalación.

RF.9.2.6.1: El lugar de instalación podrá tomar los valores 'Oficina', 'Planta', 'Laboratorio', 'Indefinido'.

RF.9.2.7: Modo de operación.

RF.9.2.7.1: El modo de operación podrá tomar los valores 'Pruebas', 'Producción', 'Indefinido'.

RF.9.2.8: Otros comentarios.

RF.9.2.8.1: Este valor será opcional.

RF.9.3: El usuario escribirá un nombre identificativo para el nuevo sensor, que podrá contener caracteres alfanuméricos y espacios.

RF.9.3.1: Ese nombre será usado siempre que el sistema deba referirse al sensor de forma visible para el usuario.

RF.9.3.2: El nombre no tendrá por qué ser único.

RF.9.4: Cada tipo de sensor podrá ofrecer una serie de valores de configuración propios como parte del proceso de conexión.

RF.9.4.1: 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.10: Los usuarios podrán ver los detalles de cualquiera de los sensores conectados.

RF.10.1: Los datos que se mostrarán para cada sensor serán los siguientes:

RF.10.1.1: Dirección IP.

RF.10.1.2: Puerto.

RF.10.1.3: Tipo de sensor.

RF.10.1.4: País de instalación.

RF.10.1.5: Ciudad de instalación.

RF.10.1.6: Lugar de instalación.

RF.10.1.6.1: El lugar de instalación podrá tomar los valores 'Oficina', 'Planta', 'Laboratorio', 'Indefinido'.

- RF.10.1.7: Modo de operación.
  - RF.10.1.7.1: El modo de operación podrá tomar los valores 'Pruebas', 'Producción', 'Indefinido'
- RF.10.1.8: Otros comentarios.
- RF.10.1.9: Nombre identificativo.
- RF.10.1.10: Cada uno de sus valores de configuración propios.
- RF.11: Los usuarios podrán eliminar sensores existentes del sistema.
  - RF.11.1: Este proceso será irreversible, de forma que, si el usuario quiere añadir el sensor de nuevo, deberá hacerlo desde cero.
  - RF.11.2: Sin embargo, el servidor remoto y el proceso modular del sensor no serán alterados por esta acción, sino que la eliminación actúa solamente en el cliente.
- RF.12: Los usuarios podrán modificar los siguientes parámetros de un sensor actualmente conectado al sistema:
  - RF.12.1: Dirección IP.
  - RF.12.2: Puerto.
  - RF.12.3: Tipo de sensor.
  - RF.12.4: País de instalación.
  - RF.12.5: Ciudad de instalación.
  - RF.12.6: Lugar de instalación.
    - RF.12.6.1: El lugar de instalación podrá tomar los valores 'Oficina', 'Planta', 'Laboratorio', 'Indefinido'.
  - RF.12.7: Modo de operación.
    - RF.12.7.1: El modo de operación podrá tomar los valores 'Pruebas', 'Producción', 'Indefinido'.
  - RF.12.8: Otros comentarios.
  - RF.12.9: Nombre identificativo.
  - RF.12.10: Cada uno de sus valores de configuración propios.
- RF.13: El sistema permitirá validar el formato de los parámetros de conexión indicados en los requisitos RF.9.2 y RF.12 antes de guardarlos.
  - RF.13.1: Aunque el formato supere la validación, el sistema no garantizará que el contenido permita comunicarse de forma efectiva con el sensor.
  - RF.13.2: Si la validación falla, el sistema mostrará un mensaje de error, y no permitirá continuar hasta que los valores tengan un formato válido.
  - RF.13.3: Si la validación falla, el usuario podrá elegir cancelar la acción en lugar de corregir el formato. En este caso, el sistema no añadirá o modificará los sensores.
- RF.14: El sistema será compatible con los siguientes tipos de sensores:
  - RF.14.1: Sensor láser lineal Gocator, de LMI Technologies.
    - RF.14.1.1: Este tipo de sensor no necesitará ningún valor de configuración propio, según se indica en RF.9.4.
- RF.15: El sistema ofrecerá la posibilidad de que los usuarios modifiquen valores de configuración:
  - RF.15.1: Estos valores persistirán entre ejecuciones del sistema.

- RF.15.2: El usuario podrá modificar el número de intentos de conexión antes de decidir que un sensor es inalcanzable.
  - RF.15.2.1: El valor por defecto para este ajuste será de 1.
- RF.15.3: El usuario podrá modificar el intervalo de tiempo en segundos entre actualizaciones de los sensores.
  - RF.15.3.1: El valor por defecto para este ajuste será de 60
- RF.16: El sistema permitirá al usuario validar el formato de los valores de configuración que lo admitan antes de guardarlos:
  - RF.16.1: El sistema comprobará que los ajustes numéricos tengan como valor un número.
  - RF.16.2: El número de intentos de conexión deberá estar entre 1 y 10.
  - RF.16.3: El intervalo de actualización deberá estar entre 10 y 600.
  - RF.16.4: Si la validación falla, el sistema mostrará un mensaje de error y no permitirá modificar los ajustes hasta que el formato sea válido.
  - RF.16.5: Si la validación falla, el usuario podrá cancelar la modificación en lugar de corregir el formato. En este caso, el sistema no modificará los ajustes.
- RF.17: El sistema permitirá restaurar los valores por defecto de la aplicación.
  - RF.17.1: Esto implica que los ajustes volverán a su valor por defecto.
  - RF.17.2: Además, todos los sensores conectados serán eliminados, dejando la aplicación lista para empezar desde cero.
- RF.18: El sistema permitirá al usuario exportar la configuración actual.
  - RF.18.1: Para ello, le permitirá elegir el directorio donde se guardará el archivo.
  - RF.18.2: El archivo exportado tendrá formato XML.
    - RF.18.2.1: El formato del archivo es el indicado en RF.26.
- RF.19: El sistema permitirá al usuario importar un archivo de configuración.
  - RF.19.1: Para ello, le permitirá elegir el archivo que quiere importar.
  - RF.19.2: El archivo para importar tendrá formato XML.
    - RF.19.2.1: El formato del archivo es el indicado en RF.26.
  - RF.19.3: Si el archivo tiene formato y contenido correctos, los ajustes actuales serán reemplazados por aquellos contenidos en el archivo.
  - RF.19.4: Si el archivo tiene errores de formato o contenido, los ajustes actuales se mantendrán sin cambios.
- RF.20: El sistema ofrecerá al usuario un sistema de Acceso Rápido.
  - RF.20.1: El Acceso Rápido podrá ser iniciado desde cualquier pantalla.
  - RF.20.2: Al iniciar el Acceso Rápido, la interacción con la pantalla actual quedará deshabilitada.
  - RF.20.3: En cualquier momento durante la interacción con el Acceso Rápido el usuario podrá salir del mismo.
    - RF.20.3.1: En este caso se cancelará cualquier operación en curso, y el usuario volverá a la pantalla en la que estaba anteriormente.
  - RF.20.4: El Acceso Rápido permitirá al usuario navegar a la pantalla de configuración desde cualquier parte del sistema.

- RF.20.5: El Acceso Rápido permitirá al usuario ver listados de sensores:
- RF.20.5.1: Antes de mostrar cualquier listado, le permitirá elegir un criterio de filtrado de entre los siguientes:
    - RF.20.5.1.1: Ver todos los sensores.
    - RF.20.5.1.2: Ver los sensores de un país concreto.
    - RF.20.5.1.3: Ver los sensores instalados en un lugar concreto.
    - RF.20.5.1.4: Ver los sensores de un tipo concreto.
    - RF.20.5.1.5: Ver los sensores trabajando en un modo concreto.
  - RF.20.5.2: Una vez elegido el criterio, el sistema mostrará la pantalla de listado con el criterio seleccionado ya aplicado.
    - RF.20.5.2.1: Los sensores se mostrarán ordenados por país, excepto si el criterio es el indicado en RF.20.5.1.2, en cuyo caso se mostrarán ordenados por tipo de sensor.
    - RF.20.5.2.2: A partir de este momento, la pantalla de listado se comportará de forma normal, como si el usuario hubiera llegado a ella sin usar el Acceso Rápido.
  - RF.20.5.3: En cualquier momento durante la selección de criterios, el usuario podrá volver atrás para cancelar la operación.
- RF.20.6: El Acceso Rápido permitirá al usuario conectar un nuevo sensor al sistema.
- RF.20.6.1: El usuario deberá introducir los siguientes datos:
    - RF.20.6.1.1: Dirección IP.
    - RF.20.6.1.2: Puerto.
    - RF.20.6.1.3: Tipo de sensor.
    - RF.20.6.1.4: País de instalación.
    - RF.20.6.1.5: Lugar de instalación.
      - RF.20.6.1.5.1: El lugar de instalación podrá tomar los valores 'Oficina', 'Planta', 'Laboratorio', 'Indefinido'.
    - RF.20.6.1.6: Modo de operación.
      - RF.20.6.1.6.1: El modo de operación podrá tomar los valores 'Pruebas', 'Producción', 'Indefinido'.
  - RF.20.6.2: El sistema hará las validaciones correspondientes sobre esos datos.
  - RF.20.6.3: Una vez completado el proceso, el sistema mostrará la pantalla de detalles para el sensor añadido.
  - RF.20.6.4: En cualquier momento del proceso, el usuario podrá volver atrás para cancelar la operación.
- RF.20.7: El Acceso Rápido permitirá al usuario modificar sensores conectados al sistema.
- RF.20.7.1: El usuario podrá modificar los siguientes datos:
    - RF.20.7.1.1: Dirección IP.
    - RF.20.7.1.2: Puerto.
    - RF.20.7.1.3: Tipo de sensor.
    - RF.20.7.1.4: País de instalación.

RF.20.7.1.5: Lugar de instalación.

RF.20.7.1.5.1: El lugar de instalación podrá tomar los valores 'Oficina', 'Planta', 'Laboratorio', 'Indefinido'.

RF.20.7.1.6: Modo de operación.

RF.20.7.1.6.1: El modo de operación podrá tomar los valores 'Pruebas', 'Producción', 'Indefinido'.

RF.20.7.2: El sistema hará las validaciones correspondientes sobre esos datos.

RF.20.7.3: Una vez completado el proceso, el sistema mostrará la pantalla de detalles para el sensor añadido.

RF.20.7.4: En cualquier momento del proceso, el usuario podrá volver atrás para cancelar la operación.

RF.20.8: El Acceso Rápido permitirá al usuario ver los detalles de un sensor conectado al sistema.

RF.20.8.1: El usuario podrá seleccionar qué sensor desea ver.

RF.20.8.2: Una vez seleccionado, el sistema mostrará la pantalla de detalles para el sensor seleccionado.

RF.20.8.3: Antes de seleccionar el sensor, el usuario podrá volver atrás para cancelar la operación.

RF.20.9: El Acceso Rápido permitirá al usuario eliminar un sensor conectado al sistema.

RF.20.9.1: El usuario podrá seleccionar qué sensor desea eliminar.

RF.20.9.2: Una vez seleccionado, el sistema eliminará el sensor y mostrará la pantalla principal.

RF.20.9.3: Antes de seleccionar el sensor, el usuario podrá volver atrás para cancelar la operación.

#### 4.2.1.1.2 Librería de configuración

RF.21: El sistema de configuración gestionará ajustes para el sistema.

RF.21.1: Los ajustes se modelan como una clave y un valor.

RF.21.2: Tanto la clave como el valor serán cadenas de texto.

RF.21.3: La clave solo podrá contener letras y guiones bajos.

RF.21.4: El valor podrá contener letras, números, espacios y símbolos excepto los siguientes: >, <, &.

RF.22: Los ajustes podrán ser globales o locales.

RF.22.1: Un ajuste global es aquel que no pertenece a ninguna categoría.

RF.22.2: Un ajuste local es aquel que está dentro de una categoría.

RF.22.3: Las categorías son agrupaciones de ajustes relacionados entre sí.

RF.23: Las categorías se identifican por un nombre que estará formado únicamente por letras y guiones bajos.

RF.24: No se permitirá que una categoría esté dentro de otra.

RF.25: El sistema de configuración utilizará archivos en formato XML.

- RF.25.1: Estos archivos tendrán un elemento raíz llamado Configuration, el cual tendrá un atributo Format con el valor 'CAv2'.
- RF.25.2: Los ajustes globales se representarán mediante un elemento XML hijo directo del elemento raíz.
- RF.25.2.1: Este elemento no tendrá ningún atributo.
- RF.25.2.2: El nombre de este elemento será la clave del ajuste.
- RF.25.2.3: El valor de este elemento será el valor del ajuste.
- RF.25.3: Las categorías se representarán mediante un elemento XML hijo directo del elemento raíz.
- RF.25.3.1: Este elemento no tendrá ningún atributo.
- RF.25.3.2: El nombre de este elemento será el nombre de la categoría.
- RF.25.4: Los ajustes locales se representarán como elementos XML hijos directos de la categoría a la que pertenecen.
- RF.25.4.1: Las características de estos elementos son las mismas que las descritas en RF.26.2.
- RF.26: Los nombres de los ajustes globales deben ser diferentes entre sí.
- RF.27: Los nombres de los ajustes de una categoría deben ser diferentes entre sí.
- RF.28: Los nombres de un ajuste global y un ajuste local pueden coincidir.
- RF.29: Los nombres de dos ajustes locales en diferentes categorías pueden coincidir.
- RF.30: Los nombres de las categorías deben ser diferentes entre sí.
- RF.31: El sistema de configuración permitirá indicar el nombre del archivo de configuración deseado.
- RF.31.1: Ese nombre de archivo deberá tener extensión XML.
- RF.31.2: Si el archivo no existe, se creará uno nuevo.
- RF.31.3: Si el archivo ya existe, se abrirá.
- RF.32: El sistema de configuración permitirá escribir ajustes.
- RF.32.1: Para ello, se deberá indicar la clave y el valor del ajuste.
- RF.32.1.1: Si el ajuste es global, la clave será el propio nombre del ajuste.
- RF.32.1.2: Si el ajuste es local, la clave será el nombre de la categoría y el nombre del ajuste separados por dos puntos (:).
- RF.32.2: Si el ajuste no existe dentro del archivo, se creará un ajuste nuevo con los datos indicados.
- RF.32.3: Si el ajuste ya existe, su valor se reemplazará por el indicado.
- RF.32.4: Si la clave del ajuste tiene formato incorrecto, se lanzará una excepción.
- RF.33: El sistema de configuración permitirá leer ajustes.
- RF.33.1: Para ello, se deberá indicar la clave del ajuste.
- RF.33.1.1: El formato de la clave será el mismo que se indica en RF.33.1.1 y RF.33.1.2.
- RF.33.2: Si el ajuste no existe dentro del archivo, se devolverá una cadena vacía.
- RF.33.3: Si el ajuste ya existe, se devolverá su valor.
- RF.33.4: Si la clave del ajuste tiene formato incorrecto, se lanzará una excepción.
- RF.34: El sistema de configuración permitirá eliminar ajustes.



RF.34.1: Para ello, se indicará la clave del ajuste.

RF.34.1.1: El formato de la clave será el mismo que se indica en RF.33.1.1 y RF.33.1.2.

RF.34.2: Si el ajuste no existe dentro del archivo, el sistema no hará nada.

RF.34.3: Si el ajuste ya existe, se eliminará.

RF.34.4: Si la clave del ajuste tiene formato incorrecto, se lanzará una excepción.

RF.35: El sistema de configuración permitirá eliminar una categoría.

RF.35.1: Para ello, se indicará el nombre de la categoría.

RF.35.2: Si la categoría no existe dentro del archivo, el sistema no hará nada.

RF.35.3: Si la categoría ya existe, se eliminará junto con todos los ajustes que contiene.

RF.35.4: Si el nombre de categoría tiene un formato incorrecto, se lanzará una excepción.

#### 4.2.1.1.3 Servidor remoto

RF.36: El servidor remoto utilizará un puerto como punto de entrada de las peticiones del cliente.

RF.36.1: El número de puerto se determinará durante el desarrollo.

RF.36.2: Si el puerto estuviera ocupado, el servidor mostrará un mensaje de error e interrumpirá su lanzamiento.

RF.37: El servidor remoto podrá descargar módulos de sensor de la red.

RF.37.1: La URL deberá ser accesible, y no requerir autenticación.

RF.37.2: La URL deberá apuntar directamente a un archivo.

RF.37.3: Si el archivo tiene formato DLL, se descargará en el directorio de ejecución.

RF.37.4: Si el archivo tiene formato ZIP, sus contenidos se extraerán en el directorio de ejecución.

RF.37.5: Si la operación tiene éxito, el servidor devolverá un mensaje de acierto.

RF.37.6: En caso contrario, el servidor devolverá un mensaje de error.

RF.38: El servidor remoto podrá cargar en memoria un módulo de sensor.

RF.38.1: El archivo que contiene dicho módulo deberá estar ubicado en el directorio de ejecución.

RF.38.2: Todas las dependencias de dicho módulo deberán estar ubicadas en el directorio de ejecución.

RF.38.3: Dicho módulo deberá estar compilado para la misma plataforma que el servidor.

RF.38.4: Si se intenta cargar un módulo que ya está cargado, la operación fallará.

RF.38.5: Si el módulo que se intenta cargar contiene un tipo que entra en conflicto con otro perteneciente a un módulo ya cargado, la operación fallará.

RF.38.6: Si el módulo está compilado para una plataforma no compatible, la operación fallará.

RF.38.7: Si el servidor no puede localizar alguna de las dependencias del módulo, la operación fallará.



- RF.38.8: Si la operación falla, el servidor devolverá un mensaje de error.
- RF.38.9: En otro caso, el servidor devolverá un mensaje de acierto.
- RF.39: El servidor remoto podrá listar los módulos cargados.
  - RF.39.1: Si no hay ningún módulo cargado, devolverá una lista vacía.
  - RF.39.2: En otro caso, devolverá una lista con los nombres de los módulos.
- RF.40: El servidor remoto podrá descargar un módulo.
  - RF.40.1: Como parte de esta operación, detendrá todos los procesos lanzados desde este módulo.
  - RF.40.2: Si el nombre de módulo no se corresponde con un módulo cargado, el servidor devolverá un mensaje de error.
  - RF.40.3: En otro caso, devolverá un mensaje de acierto.
- RF.41: El servidor remoto podrá instanciar un proceso nuevo.
  - RF.41.1: El módulo que contiene el proceso deberá estar cargado en memoria.
  - RF.41.2: El tipo correspondiente al proceso deberá implementar una interfaz común.
    - RF.41.2.1: Esta interfaz incluirá métodos para cada una de las fases del ciclo de vida de los procesos:
      - RF.41.2.1.1: Inicialización
      - RF.41.2.1.2: Operación
      - RF.41.2.1.3: Mensaje del usuario
      - RF.41.2.1.4: Finalización
  - RF.41.3: Si el proceso que se intenta instanciar ya está instanciado, la operación fallará.
  - RF.41.4: Si el proceso que se intenta instanciar tiene el mismo nombre que un proceso ya instanciado (aunque provenga de un módulo diferente), la operación fallará.
  - RF.41.5: Si la operación falla, el servidor devolverá un mensaje de error.
  - RF.41.6: En otro caso, el servidor devolverá un mensaje de acierto.
  - RF.41.7: Si un proceso puede instanciarse correctamente, su ejecución empezará de inmediato.
- RF.42: El servidor remoto podrá listar los procesos en ejecución.
  - RF.42.1: Si no hay ningún proceso en ejecución, devolverá una lista vacía.
  - RF.42.2: En otro caso, devolverá una lista de nombres de proceso, asociados al módulo del que provienen.
- RF.43: El servidor remoto podrá enviar mensajes a un proceso.
  - RF.43.1: Dicho proceso deberá haber sido instanciado previamente.
  - RF.43.2: El mensaje se añadirá a la cola de mensajes pendientes del proceso.
  - RF.43.3: Si la cola está llena, se eliminarán los mensajes más antiguos para hacer sitio a los nuevos.
  - RF.43.4: Si la operación se completa correctamente, el sistema devolverá un mensaje de acierto.
  - RF.43.5: En otro caso, el sistema devolverá un mensaje de error.
- RF.44: El servidor remoto podrá pedir resultados a un proceso.

- RF.44.1: Dicho proceso deberá haber sido instanciado previamente.
- RF.44.2: Si el proceso tiene algún resultado, el servidor lo devolverá.
- RF.44.3: En otro caso, devolverá una lista vacía.
- RF.45: El servidor remoto gestionará el ciclo de vida de los procesos.
  - RF.45.1: Al instanciarse, un proceso podrá ejecutar código propio.
  - RF.45.2: Al comienzo de cada ciclo de iteración, un proceso podrá ejecutar código propio.
  - RF.45.3: Después de cada ciclo de iteración, el servidor comprobará si hay mensajes pendientes para el proceso.
    - RF.45.3.1: Si los hay, el proceso los recibirá y podrá ejecutar código propio.
    - RF.45.3.2: En otro caso, el proceso no será alertado.
  - RF.45.4: Tras esta comprobación, el servidor pondrá el proceso en espera durante el tiempo indicado.
    - RF.45.4.1: Si este tiempo es superior a un minuto, el servidor lo dividirá en fracciones de un minuto.
    - RF.45.4.2: Entre cada fracción, el servidor comprobará si hay mensajes pendientes, según se indica en RF.45.3.
  - RF.45.5: Una vez terminado el tiempo de espera, el servidor iniciará un nuevo ciclo de iteración, según se indica en RF.45.2.
  - RF.45.6: Si el proceso ha terminado su trabajo, el servidor iniciará la finalización en lugar de ejecutar otra iteración.
    - RF.45.6.1: Como parte de la finalización, el proceso podrá ejecutar código propio.
    - RF.45.6.2: Una vez que comienza la finalización, un proceso no puede arrepentirse y solicitar más iteraciones.
    - RF.45.6.3: Todos los mensajes y resultados pendientes se descartarán al finalizar un proceso.
- RF.46: Los procesos podrán modificar sus parámetros de funcionamiento en cualquier momento en el que el servidor les permita ejecutar código propio.
  - RF.46.1: Los procesos podrán modificar su intervalo de iteración.
    - RF.46.1.1: El nuevo intervalo tendrá efecto a partir de la próxima iteración.
    - RF.46.1.2: Si el intervalo es menor a un segundo, el servidor ignorará la petición y lo establecerá en un segundo.
    - RF.46.1.3: Si el intervalo es mayor a una hora, el servidor ignorará la petición y lo establecerá en una hora.
  - RF.46.2: Los procesos podrán emitir resultados.
    - RF.46.2.1: Los resultados se insertarán en la cola de resultados pendientes.
    - RF.46.2.2: Si la cola está llena, se eliminarán los resultados más antiguos para hacer sitio a los nuevos.
  - RF.46.3: Los procesos pueden solicitar su finalización.
    - RF.46.3.1: En este caso, el servidor garantiza que no ejecutará ninguna otra iteración.

#### 4.2.1.2 Requisitos no funcionales

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

RNF.1.1: 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 remotos tendrán instalada la plataforma .NET Framework 4.6.1

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

RNF.4: La máquina local y las máquinas remotas estarán conectadas a la misma red local, o en su defecto, la máquina local deberá poder conectarse a las máquinas remotas.

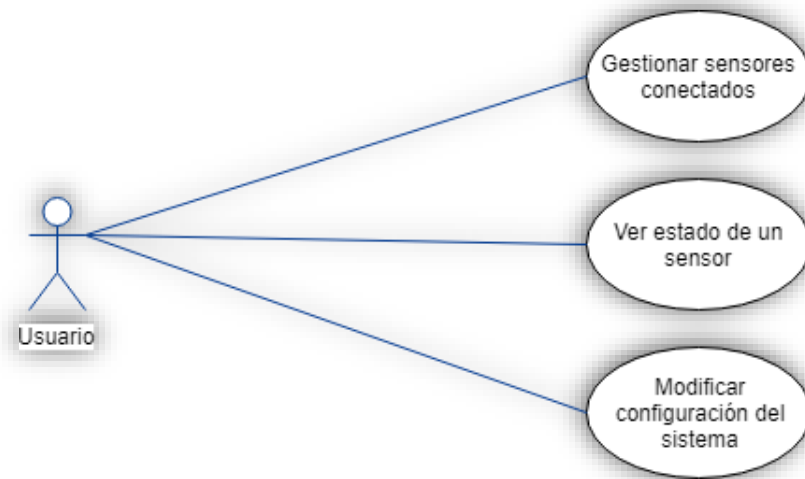
## 4.2.2 Identificación de Actores del Sistema

A continuación se identificarán los actores del sistema, tanto primarios como secundarios:

Aplicación local	
Actores primarios	Actores secundarios
<ul style="list-style-type: none"><li>▪ <b>Usuario:</b> Representa a las personas que interactúan con la aplicación, haciendo uso de cualquiera de sus funciones.</li></ul>	<ul style="list-style-type: none"><li>▪ <b>Servidores remotos:</b> Representa a cada uno de los servidores remotos con los que la aplicación se comunica para conseguir los datos necesarios.</li></ul>
Servidores remotos	
Actores primarios	Actores secundarios
<ul style="list-style-type: none"><li>▪ <b>Aplicación local:</b> Representa a la aplicación local, que solicita datos en respuesta a una orden del usuario.</li></ul>	

### 4.2.3 Especificación de Casos de Uso

En esta sección se incluye el diagrama de casos de uso de la aplicación, y un análisis de cada uno de ellos:



*Figura 4.1. Diagrama de casos de uso*

Nombre del Caso de Uso
Gestionar sensores conectados
Descripción
<p>El usuario debe poder gestionar los sensores que ha conectado. Esto incluye las siguientes acciones:</p> <ul style="list-style-type: none"><li>▪ Ver distribución de los sensores conectados en un mapa.</li><li>▪ Ver, filtrar y ordenar un listado de los sensores conectados.</li><li>▪ Conectar nuevos sensores al sistema.</li><li>▪ Modificar alguno de los sensores conectados.</li><li>▪ Desconectar alguno de los sensores conectados.</li></ul>

Nombre del Caso de Uso
Ver estado de un sensor
Descripción
El usuario puede acceder a los detalles de uno de los sensores conectados. La aplicación le mostrará los parámetros de ese sensor, su estado, y la información recibida del mismo (a no ser que el sensor esté desconectado).

Nombre del Caso de Uso
Modificar la configuración del sistema
Descripción
El usuario puede acceder a la configuración. De forma visual (sin editar manualmente ningún archivo) puede cambiar los parámetros que se le indique.
Al guardar los cambios, la aplicación se reiniciará, y a partir de ese momento empezará a usar los nuevos valores de configuración.

## 4.3 Identificación de los Subsistemas en la Fase de Análisis

### 4.3.1 Descripción de los Subsistemas

A continuación se pasará a enumerar los subsistemas que componen la aplicación, junto con una descripción de su propósito:

- **Aplicación local:** Representa la aplicación local e incluye los componentes de interfaz de usuario necesarios para que el usuario perciba e interactúe con los datos.
- **Servidor remoto:** Representa los contenedores de procesos modulares que se despliegan en las máquinas remotas. Además de recibir peticiones desde la aplicación local y transmitir datos de respuesta, también gestionan el ciclo de vida de los procesos de sensor.

### 4.3.2 Descripción de los Interfaces entre Subsistemas

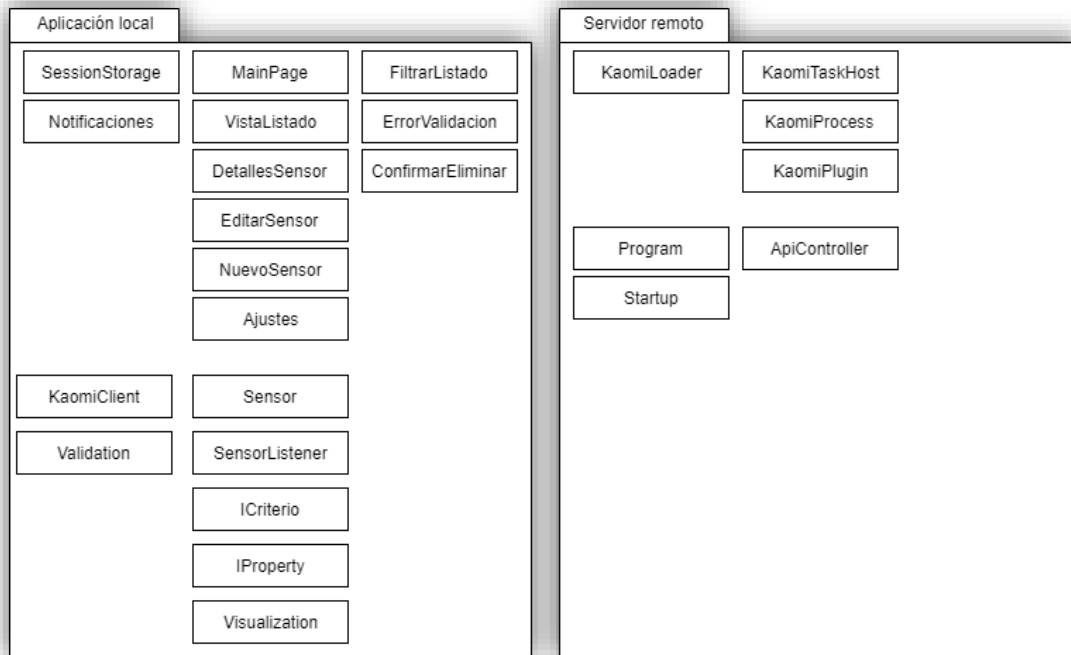
Las comunicaciones entre subsistemas identificadas son las siguientes:

- **Entre la aplicación local y los servidores remotos:** Se comunican a través de la red mediante peticiones REST sobre HTTP o HTTPS.

## 4.4 Diagrama de Clases Preliminar del Análisis

### 4.4.1 Diagrama de Clases

Esta sección muestra un resumen del diagrama de clases preliminar del proyecto:



*Figura 4.2. Diagrama de clases preliminar*



## 4.4.2 Descripción de las Clases

### 4.4.2.1 Aplicación local

Nombre de la Clase
SessionStorage
Descripción
Clase estática que mantiene la lista de sensores conectados al sistema.
Responsabilidades
<ul style="list-style-type: none"> <li>Almacén único de sensores conectados.</li> <li>Añadir y eliminar sensores según el usuario los conecte o desconecte.</li> </ul>
Atributos Propuestos
<b>Sensores:</b> Colección de sensores.
Métodos Propuestos
<b>AddSensor:</b> Añade un sensor a la colección.
<b>RemoveSensor:</b> Elimina un sensor de la colección.
<b>ClearSensores:</b> Elimina todos los sensores conectados.

Nombre de la Clase
Notificaciones
Descripción
Clase estática que se encarga de enviar notificaciones de escritorio cuando los sensores cambian de estado.
Responsabilidades
<ul style="list-style-type: none"> <li>Crear y emitir notificaciones de escritorio.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<b>Mostrar:</b> Crea y muestra una notificación de escritorio.

Nombre de la Clase
MainPage
Descripción
Código correspondiente a la pantalla principal de la aplicación.
Responsabilidades
<ul style="list-style-type: none"> <li>Mostrar distribución geográfica de sensores.</li> <li>Actualizar periódicamente los iconos de mapa según el estado de los sensores.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<b>UpdateMapIcons:</b> Actualiza el aspecto de los iconos en el mapa. Este método será llamado periódicamente.

**MapElementClick:** Como respuesta a la pulsación sobre uno de los iconos de mapa, lleva al usuario al listado de sensores del país correspondiente.

Nombre de la Clase
VistaListado
Descripción
Código correspondiente a la pantalla de listado de sensores.
Responsabilidades
<ul style="list-style-type: none"> <li>Mostrar listado de sensores.</li> <li>Actualizar el listado en función de los cambios en criterios de filtrado.</li> </ul>
Atributos Propuestos
<b>Visualización:</b> Contiene los criterios activos actualmente.
Métodos Propuestos
<b>UpdateList:</b> Actualiza el listado como respuesta a un cambio de criterios.

Nombre de la Clase
DetallesSensor
Descripción
Código correspondiente a la pantalla de detalles de un sensor.
Responsabilidades
<ul style="list-style-type: none"> <li>Mostrar información sobre un sensor.</li> <li>Actualizar periódicamente la información del sensor.</li> </ul>
Atributos Propuestos
<b>Sensor:</b> Sensor del que se están viendo los detalles.
Métodos Propuestos
<b>UpdateProperties:</b> Actualiza la información del sensor.

Nombre de la Clase
EditarSensor
Descripción
Código correspondiente a la pantalla de modificación de un sensor.
Responsabilidades
<ul style="list-style-type: none"> <li>Modificar los datos de un sensor ya conectado.</li> </ul>
Atributos Propuestos
<b>Sensor:</b> Sensor que está siendo modificado.
Métodos Propuestos
<b>SaveSensor:</b> Confirma las modificaciones del sensor.

Nombre de la Clase
NuevoSensor
Descripción

Código correspondiente a la pantalla de conexión de un nuevo sensor.
Responsabilidades
<ul style="list-style-type: none"> <li>Pedir y validar los datos correspondientes a un nuevo sensor.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<b>AddSensor:</b> Crea y guarda un sensor con los datos proporcionados por el usuario.

Nombre de la Clase
Ajustes
Descripción
Código correspondiente a la pantalla de ajustes del sistema.
Responsabilidades
<ul style="list-style-type: none"> <li>Mostrar y modificar los ajustes del sistema.</li> <li>Permitir devolver la aplicación a la configuración por defecto.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<b>PopulateSettings:</b> Muestra los valores actuales de los ajustes.
<b>SaveSettings:</b> Modifica los valores de los ajustes.
<b>RestoreSettings:</b> Devuelve los ajustes a sus valores por defecto.

Nombre de la Clase
FiltrarListado
Descripción
Código correspondiente al diálogo de filtros de listado.
Responsabilidades
<ul style="list-style-type: none"> <li>Mostrar y modificar los criterios de filtrado.</li> <li>Eliminar todos los filtros activos.</li> </ul>
Atributos Propuestos
<b>Visualización:</b> Contiene los criterios que se van a aplicar al listado.
Métodos Propuestos
<b>ApplyFilters:</b> Aplica los nuevos criterios al listado.
<b>RemoveFilters:</b> Elimina todos los filtros activos.

Nombre de la Clase
ErrorValidacion
Descripción
Código correspondiente al diálogo que informa de un error de validación.
Responsabilidades
<ul style="list-style-type: none"> <li>Informar al usuario de que un campo tiene formato incorrecto.</li> </ul>

Atributos Propuestos
<b>Field:</b> Nombre del campo con formato incorrecto.
Métodos Propuestos

Nombre de la Clase
ConfirmarEliminar
Descripción
Código correspondiente al diálogo que solicita confirmación antes de eliminar un sensor.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Recabar confirmación del usuario antes de eliminar un sensor.</li> </ul>
Atributos Propuestos
Métodos Propuestos

Nombre de la Clase
KaomiClient
Descripción
Clase que se comunica con un servidor remoto.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Establecer comunicación con un proceso sensor remoto.</li> <li>▪ Solicitar resultados al proceso sensor remoto.</li> </ul>
Atributos Propuestos
<b>IP:</b> Dirección IP del servidor remoto.
<b>Puerto:</b> Puerto del servidor remoto.
Métodos Propuestos
<b>Connect:</b> Se conecta al servidor remoto.
<b>LatestResults:</b> Consigue el último resultado producido por el servidor remoto.

Nombre de la Clase
Validation
Descripción
Implementa lógica de validación para la interfaz de usuario.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Validar el formato de cadenas de texto.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<b>ValidateIP:</b> Comprueba si el texto representa una dirección IP válida.
<b>ValidatePort:</b> Comprueba si el texto representa un puerto válido.

Nombre de la Clase
Sensor
Descripción
Modela un sensor dentro del sistema.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Guardar los atributos y propiedades de un sensor.</li> <li>▪ Determinar el estado del sensor en función de sus propiedades.</li> </ul>
Atributos Propuestos
<b>Código</b> ID, Nombre, IP, Puerto, Tipo, Ubicación, País, Ciudad, Modo, Comentarios:
Atributos del sensor.
<b>Propiedades:</b> Almacena las propiedades recibidas desde el servidor remoto.
Métodos Propuestos
<b>GetStatus:</b> Determina el estado del sensor en función de sus propiedades actuales.

Nombre de la Clase
SensorListener
Descripción
Implementa el patrón Observer para responder a cambios de estado en un sensor.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Hacer una llamada cuando un sensor cambie de estado.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<b>OnStatusChanged:</b> Método llamado cuando el sensor cambia de estado.

Nombre de la Clase
ICriterio
Descripción
Interfaz común a cada uno de los criterios de filtrado que pueden aplicarse.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Indicar cómo un atributo concreto se va a filtrar.</li> </ul>
Atributos Propuestos
<b>Predicado:</b> Expresión lógica que determina si un atributo supera el filtro.
Métodos Propuestos

Nombre de la Clase
Property
Descripción
Clase que modela las propiedades recibidas desde un servidor remoto.
Responsabilidades

<ul style="list-style-type: none"> <li>▪ Guardar el nombre y el valor de una propiedad.</li> </ul>
Atributos Propuestos
<b>Key:</b> Nombre de la propiedad.
<b>Value:</b> Valor de la propiedad.
Métodos Propuestos

Nombre de la Clase
Visualization
Descripción
Agrupar un criterio para cada atributo posible, creando una instancia completa del filtro.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Guardar los criterios correspondientes a un filtro.</li> <li>▪ Determinar si los sensores superan el filtro.</li> </ul>
Atributos Propuestos
<b>Type, Country, Location, Mode:</b> Criterios para cada atributo posible.
Métodos Propuestos
<b>Apply:</b> Aplica el filtro a un sensor y comprueba si lo cumple.

#### 4.4.2.2 Servidor remoto

Nombre de la Clase
KaomiLoader
Descripción
Administra la carga por reflexión de procesos de sensor en los servidores remotos.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Obtener y preparar ensamblados para su carga en memoria.</li> <li>▪ Instanciar tipos concretos dentro de esos ensamblados.</li> <li>▪ Solicitar resultados a los procesos instanciados.</li> </ul>
Atributos Propuestos
<b>Assemblies:</b> Colección de ensamblados cargados en memoria.
<b>TaskHosts:</b> Colección de procesos en ejecución.
Métodos Propuestos
<b>LoadAssembly:</b> Carga un ensamblado en memoria.
<b>InstanceProcess:</b> Instancia un proceso desde un ensamblado previamente cargado.
<b>GetResults:</b> Solicita resultados a un proceso concreto.

Nombre de la Clase
KaomiTaskHost
Descripción
Gestiona el funcionamiento y el ciclo de vida de un proceso dentro del servidor.

Responsabilidades
<ul style="list-style-type: none"> <li>▪ Mantener el proceso iterando mientras sea necesario.</li> <li>▪ Enviar resultados cuando sean solicitados.</li> </ul>
Atributos Propuestos
<b>Results:</b> Cola de resultados pendientes de envío.
Métodos Propuestos
<b>RunProcess:</b> Pone en ejecución el proceso.
<b>WaitIteration:</b> Espera el intervalo de iteración correspondiente.

Nombre de la Clase
KaomiProcess
Descripción
Clase abstracta que modela cada uno de los procesos que pueden ejecutarse.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Implementar una interfaz común para todos los procesos.</li> <li>▪ Contener la lógica de negocio correspondiente a la tarea deseada.</li> </ul>
Atributos Propuestos
<b>IterationDelay:</b> Tiempo de espera entre iteraciones.
Métodos Propuestos
<b>OnInitialize:</b> Se ejecuta al cargar el proceso en memoria.
<b>OnIteration:</b> Se ejecuta una vez por cada iteración.
<b>OnFinalize:</b> Se ejecuta al descargar el proceso de memoria.

Nombre de la Clase
KaomiPlugin
Descripción
Clase abstracta que permite agrupar funcionalidad común de manera que los procesos puedan utilizarla sin necesidad de implementarla por sí mismos.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Ofrecer una interfaz común para solicitar y utilizar plugins.</li> </ul>
Atributos Propuestos
Métodos Propuestos
<b>Initialize:</b> Ejecuta la lógica de inicialización del plugin, si fuera necesaria.

Nombre de la Clase
Program
Descripción
Clase estática que representa el punto de entrada para la ejecución del servidor.
Responsabilidades
<ul style="list-style-type: none"> <li>▪ Iniciar la ejecución del servidor remoto.</li> </ul>

Atributos Propuestos
Métodos Propuestos
<b>Main:</b> Punto de entrada para la ejecución.

Nombre de la Clase
Startup
Descripción
Configura los servicios y el middleware de ASP.NET durante la inicialización.
Responsabilidades
<ul style="list-style-type: none"><li>▪ Configurar el servidor remoto.</li></ul>
Atributos Propuestos
Métodos Propuestos
<b>Configure:</b> Modifica la configuración del servidor.

Nombre de la Clase
ApiController
Descripción
Implementa los endpoints para las peticiones REST entrantes.
Responsabilidades
<ul style="list-style-type: none"><li>▪ Recibir peticiones desde la interfaz de usuario.</li><li>▪ Procesar datos y enviar mensajes de respuesta.</li></ul>
Atributos Propuestos
Métodos Propuestos
<b>LoadAssembly:</b> Carga un ensamblado en memoria. <b>InstanceProcess:</b> Instancia un proceso desde un ensamblado previamente cargado. <b>GetResults:</b> Solicita resultados a un proceso concreto.



## 4.5 Análisis de Casos de Uso y Escenarios

En esta sección se detallarán los escenarios de uso identificados para cada uno de los casos de uso del sistema.

### 4.5.1 Gestionar sensores conectados

Ver distribución de sensores	
Precondiciones	Debe haber al menos un sensor conectado al sistema.
Poscondiciones	Ninguna
Actores	Iniciado y terminado por cualquier usuario
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ En la pantalla principal podrá ver iconos en un mapa, mostrando la distribución geográfica de los sensores conectados.</li> <li>▪ El color del icono le informará del estado de esos sensores.</li> </ul>
Variaciones (escenarios secundarios)	
Excepciones	
Notas	

Ver listado de sensores	
Precondiciones	Debe haber al menos un sensor conectado al sistema
Poscondiciones	Ninguna
Actores	Iniciado y finalizado por cualquier usuario
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará al listado de todos los sensores.</li> <li>▪ El sistema le mostrará una lista con todos los sensores conectados al sistema.</li> <li>▪ Cada sensor tendrá un icono indicando su estado.</li> </ul>
Variaciones (escenarios secundarios)	
Excepciones	
Notas	

Filtrar listado de sensores	
Precondiciones	Debe haber al menos dos sensores conectados al sistema.
Poscondiciones	Ninguna
Actores	Iniciado y finalizado por cualquier usuario.
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará al listado de todos los sensores.</li> <li>▪ Abrirá el cuadro de diálogo de filtros y los modificará de forma que solo parte de los sensores lo cumplan.</li> <li>▪ Al aplicar el filtro, solo verá aquellos sensores que cumplen las condiciones indicadas.</li> </ul>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 1:</b> Las condiciones no son cumplidas por ningún sensor conectado. <ul style="list-style-type: none"> <li>○ Al aplicar los filtros, el usuario verá una lista vacía.</li> </ul> </li> <li>▪ <b>Escenario Alternativo 2:</b> Las condiciones son cumplidas por todos los sensores. <ul style="list-style-type: none"> <li>○ Al aplicar los filtros, el usuario verá todos los sensores en la lista.</li> </ul> </li> </ul>
Excepciones	
Notas	

Modificar filtros de sensores	
Precondiciones	Debe haber al menos dos sensores conectados al sistema.
Poscondiciones	Ninguna
Actores	Iniciado y finalizado por cualquier usuario.
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará al listado de todos los sensores.</li> <li>▪ Abrirá el cuadro de diálogo de filtros y los modificará de forma que solo parte de los sensores lo cumplan.</li> <li>▪ Al aplicar el filtro, solo verá aquellos sensores que cumplen las condiciones indicadas.</li> <li>▪ Volverá a abrir el diálogo de filtros y los modificará por segunda vez.</li> <li>▪ Al aplicar el nuevo filtro, verá un listado actualizado con los sensores que cumplen el nuevo criterio.</li> </ul>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 1:</b> Las nuevas condiciones no son cumplidas por ningún sensor conectado. <ul style="list-style-type: none"> <li>○ Al aplicar los filtros, el usuario verá una lista vacía.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 2:</b> Las nuevas condiciones son cumplidas por todos los sensores. <ul style="list-style-type: none"> <li>○ Al aplicar los filtros, el usuario verá todos los sensores en la lista.</li> </ul> </li> </ul>
Excepciones	
Notas	

Añadir un nuevo sensor	
Precondiciones	La máquina remota debe tener un servidor desplegado y en ejecución.
Poscondiciones	El sistema mostrará un sensor más que antes.
Actores	Iniciado y finalizado por cualquier usuario.
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará a la pantalla de conexión de nuevos sensores.</li> <li>▪ El sistema le guiará por los datos necesarios para conectar el sensor.</li> <li>▪ Al terminar, el sistema le devolverá a la pantalla principal.</li> <li>▪ Si el usuario navega al listado de todos los sensores, verá el sensor que acaba de añadir.</li> </ul>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 1:</b> Alguno de los datos introducidos tiene formato incorrecto. <ul style="list-style-type: none"> <li>○ El sistema notificará el error al usuario y no le permitirá continuar hasta que corrija el fallo.</li> </ul> </li> <li>▪ <b>Escenario Alternativo 2:</b> El usuario cancela la conexión del sensor. <ul style="list-style-type: none"> <li>○ El sistema le devolverá a la pantalla principal y no hará cambios en la lista de sensores activos.</li> </ul> </li> </ul>
Excepciones	
Notas	

Modificar sensor existente	
Precondiciones	Debe haber al menos un sensor conectado al sistema.
Poscondiciones	El sensor modificado usará los nuevos datos.
Actores	Iniciado y finalizado por cualquier usuario.
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará al listado de todos los sensores.</li> <li>▪ Desde el listado, navegará a los detalles de un sensor.</li> <li>▪ Desde esa pantalla, navegará a la pantalla de modificación</li> </ul>

	<p>de datos.</p> <ul style="list-style-type: none"> <li>▪ Modificará alguno de los datos disponibles y guardará los cambios.</li> <li>▪ El sistema de llevará de vuelta a la pantalla de detalles, donde podrá comprobar que los cambios se han guardado.</li> </ul>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 1:</b> El usuario cancela la modificación. <ul style="list-style-type: none"> <li>○ El usuario volverá a la pantalla de detalles y podrá comprobar que estos no han cambiado.</li> </ul> </li> <li>▪ <b>Escenario Alternativo 2:</b> El usuario guarda los cambios, pero todos los campos tienen los mismos valores que antes. <ul style="list-style-type: none"> <li>○ El usuario volverá a la pantalla de detalles y podrá comprobar que estos no han cambiado.</li> </ul> </li> </ul>
Excepciones	
Notas	

Eliminar un sensor	
Precondiciones	Debe haber al menos un sensor conectado al sistema.
Poscondiciones	El sensor eliminado dejará de estar visible.
Actores	Iniciado y finalizado por cualquier usuario.
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará al listado de todos los sensores.</li> <li>▪ Desde el listado, navegará a los detalles de un sensor.</li> <li>▪ Desde esa pantalla, pulsará sobre la opción de eliminar el sensor.</li> <li>▪ Al confirmar la eliminación, el sistema le llevará a la pantalla principal.</li> <li>▪ Si accede al listado de todos los sensores, podrá comprobar que el sensor eliminado ya no aparece.</li> </ul>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 1:</b> El usuario cancela la eliminación. <ul style="list-style-type: none"> <li>○ El sistema le dejará en la pantalla de detalles, y el usuario podrá comprobar que el sensor sigue conectado.</li> </ul> </li> </ul>
Excepciones	
Notas	

## 4.5.2 Ver estado de un sensor

Ver estado de un sensor	
Precondiciones	Debe haber al menos un sensor conectado al sistema.
Poscondiciones	Ninguna
Actores	Iniciado y finalizado por cualquier usuario.
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará al listado de todos los sensores.</li> <li>▪ Desde el listado, navegará a la pantalla de detalles de uno de los sensores.</li> <li>▪ El sistema le mostrará la información disponible sobre ese sensor, incluyendo las propiedades recibidas desde el servidor remoto.</li> </ul>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 1:</b> El sensor está desconectado. <ul style="list-style-type: none"> <li>○ El sistema no mostrará las propiedades recibidas, pero sí el resto de datos.</li> </ul> </li> </ul>
Excepciones	
Notas	

## 4.5.3 Modificar la configuración del sistema

Modificar la configuración del sistema	
Precondiciones	Ninguna
Poscondiciones	Ninguna
Actores	Iniciado y finalizado por cualquier usuario.
Descripción	<p>El usuario:</p> <ul style="list-style-type: none"> <li>▪ Abrirá la aplicación.</li> <li>▪ Desde la pantalla principal navegará a la pantalla de configuración.</li> <li>▪ Modificará el valor de alguno de los campos.</li> <li>▪ Al guardar los cambios, la aplicación se reiniciará.</li> <li>▪ Si vuelve a la pantalla de configuración, podrá comprobar que el nuevo valor se muestra.</li> </ul>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> <li>▪ <b>Escenario Alternativo 1:</b> El valor introducido tiene formato incorrecto o está fuera de rango. <ul style="list-style-type: none"> <li>○ El sistema informará del error y no permitirá guardar los cambios hasta que se solucione.</li> </ul> </li> <li>▪ <b>Escenario Alternativo 2:</b> El usuario cancela los cambios.</li> </ul>

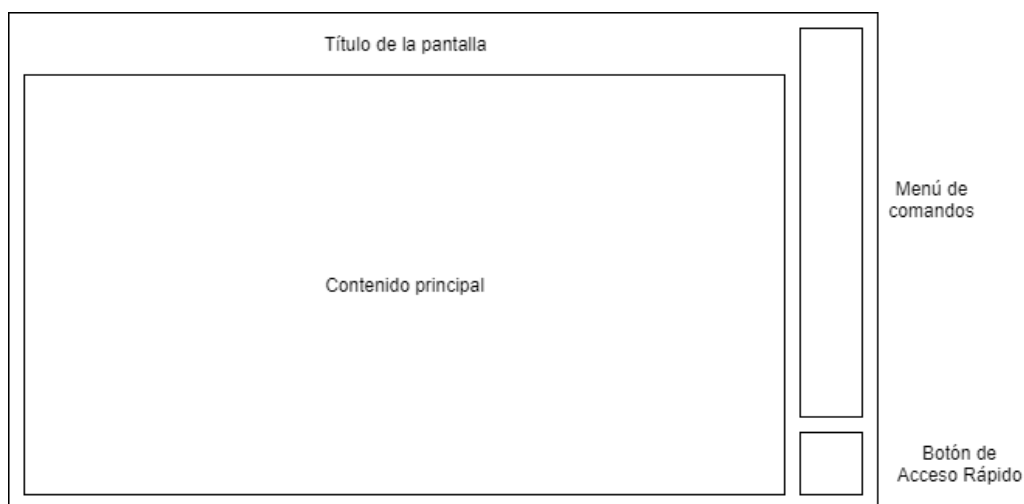
	<ul style="list-style-type: none"><li>○ El sistema le devolverá a la pantalla principal. Si el usuario vuelve a la pantalla de configuración, verá que el valor anterior se mantiene.</li><li>▪ <b>Escenario Alternativo 3:</b> El usuario guarda los cambios, pero todos los campos tienen el valor sin modificar.<ul style="list-style-type: none"><li>○ El sistema reiniciará la aplicación. Si el usuario vuelve a la pantalla de configuración, verá que los valores originales se mantienen.</li></ul></li></ul>
Excepciones	
Notas	

## 4.6 Análisis de Interfaces de Usuario

Esta sección tratará sobre la interfaz de usuario de la aplicación local, ya que es la única parte del sistema con la que los usuarios van a interactuar. El resto de componentes no tienen una interfaz de usuario como tal, ya que su uso se hace de forma automática sin intervención del usuario.

### 4.6.1 Descripción de la Interfaz

Todas las pantallas de la aplicación siguen el mismo esquema básico de estructura, que se muestra a continuación:



*Figura 4.3. Esquema de la interfaz de usuario*

Cabe destacar que el menú de comandos cambiará en función de la pantalla en la que se encuentre el usuario, pero manteniendo los siguientes principios:

- Si el usuario puede cancelar o volver atrás, esa será la primera opción (colocada más cerca del borde superior).
- Las demás acciones se dispondrán a continuación, formando una columna de botones.
- El máximo número de acciones por pantalla será de cinco, sin incluir el acceso rápido.
- Los cuadros de diálogo que se muestren aparecerán centrados en la pantalla, tanto vertical como horizontalmente.

El aspecto visual de la aplicación se ha diseñado de acuerdo con la Guía de Estilo Flushing.

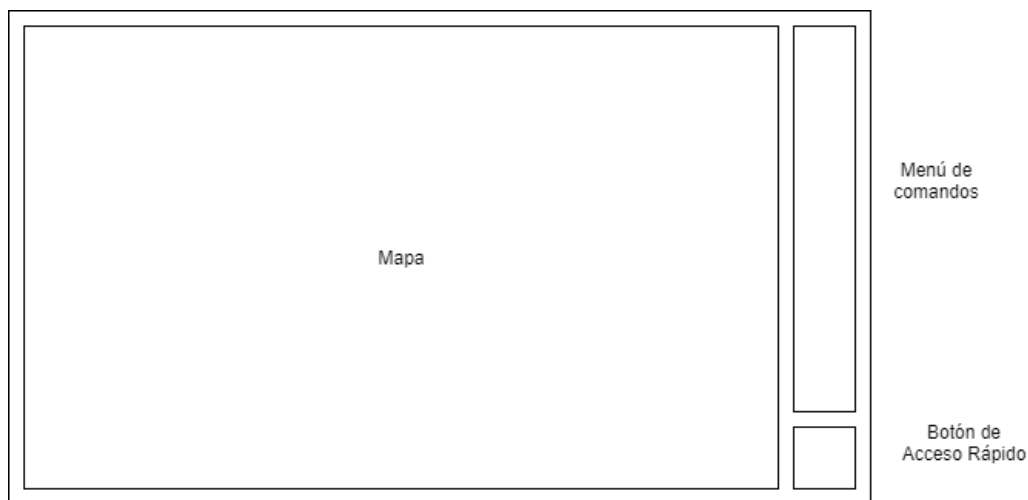
La Guía de Estilo Flushing es un documento creado por el propio autor de este proyecto, que propone una serie de guías y recomendaciones sobre los siguientes aspectos de un proyecto:

- Aspecto visual, indicando distribución de elementos, tipos de letra, y colores.
- Lenguaje, buscando un contenido normativo que se pueda aplicar a todos los registros lingüísticos.
- Comportamiento, creando modelos de navegación comunes.

Esta guía de estilo es aplicable a documentos de texto, aplicaciones de escritorio, y páginas web, y sus detalles específicos pueden encontrarse en el apartado de 'referencias' al final de este documento.

A continuación se incluyen los prototipos para cada una de las pantallas de la aplicación:

### 4.6.1.1 Pantalla principal



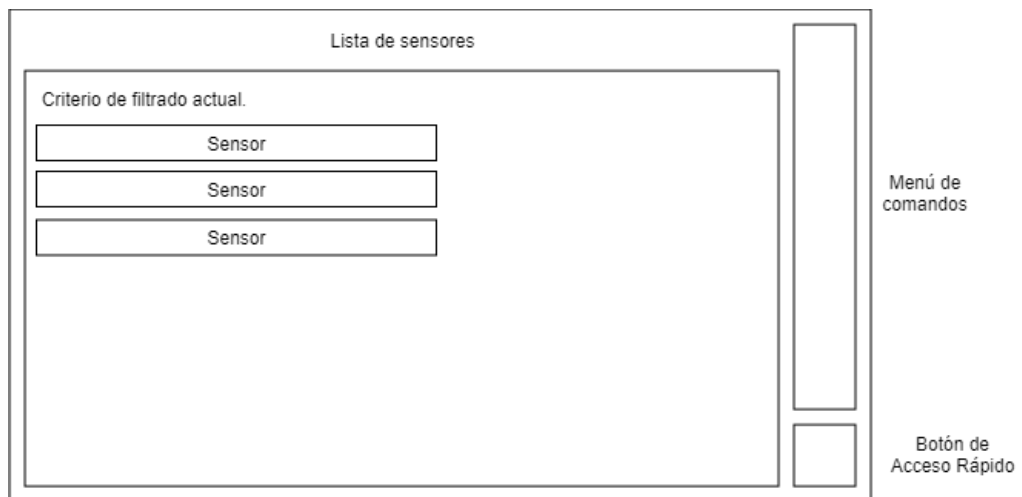
*Figura 4.4. Esquema de la pantalla principal*

La pantalla principal mostrará una distribución geográfica de los sensores conectados al sistema, además de las siguientes acciones:

- Navegar a la pantalla de conexión de un nuevo sensor.
- Navegar al listado de todos los sensores conectados.
- Navegar a la configuración del sistema.



#### 4.6.1.2 Pantalla de listado

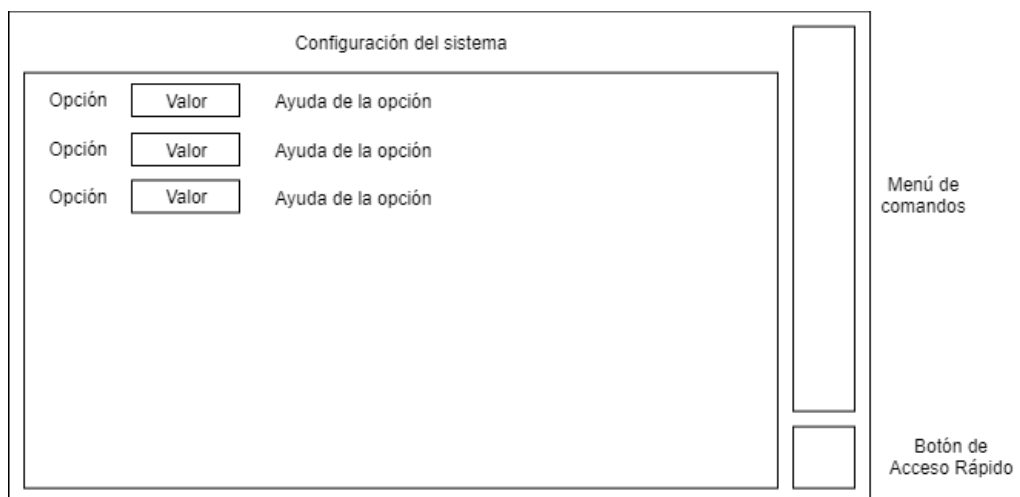


*Figura 4.5. Esquema de la pantalla de listado*

Esta pantalla muestra una lista de sensores de acuerdo a un criterio configurable. Dispone de las siguientes acciones:

- Volver atrás.
- Abrir el diálogo de filtros.

#### 4.6.1.3 Pantalla de configuración



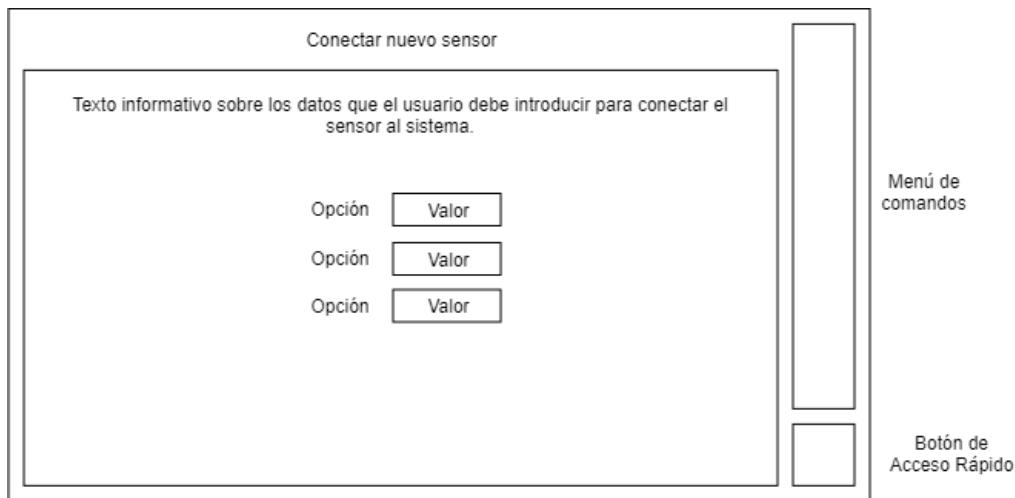
*Figura 4.6. Esquema de la pantalla de configuración*

Esta pantalla permite modificar los ajustes del sistema, e incluye las siguientes acciones:

- Volver atrás sin guardar los cambios.
- Guardar los cambios.
- Exportar la configuración a un archivo.

- Importar la configuración desde un archivo.

#### 4.6.1.4 Pantalla de conexión de un sensor nuevo

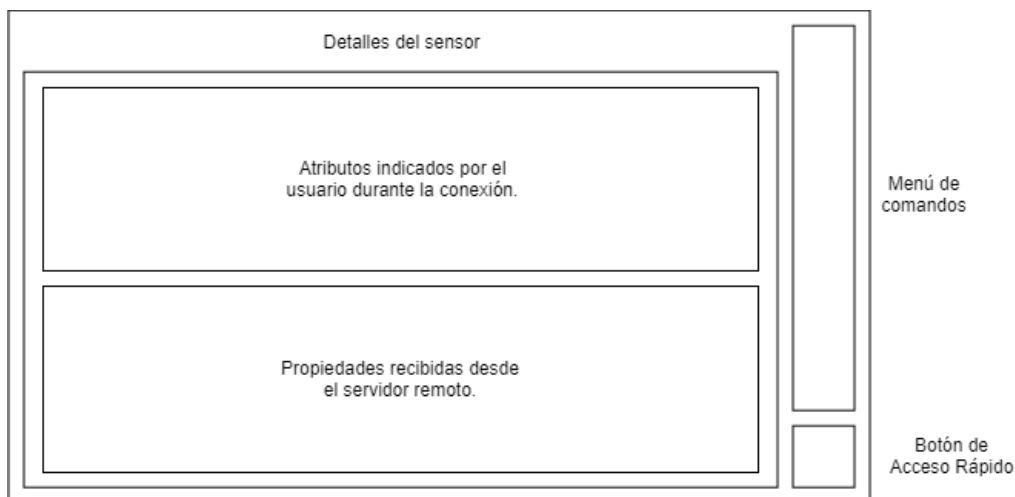


*Figura 4.7. Esquema de la pantalla de conexión de un sensor*

Esta pantalla permite al usuario indicar los datos necesarios para conectar un nuevo sensor al sistema. Ofrece las siguientes acciones:

- Cancelar la operación y volver atrás.
- Añadir el sensor al sistema.

#### 4.6.1.5 Pantalla de detalles de un sensor



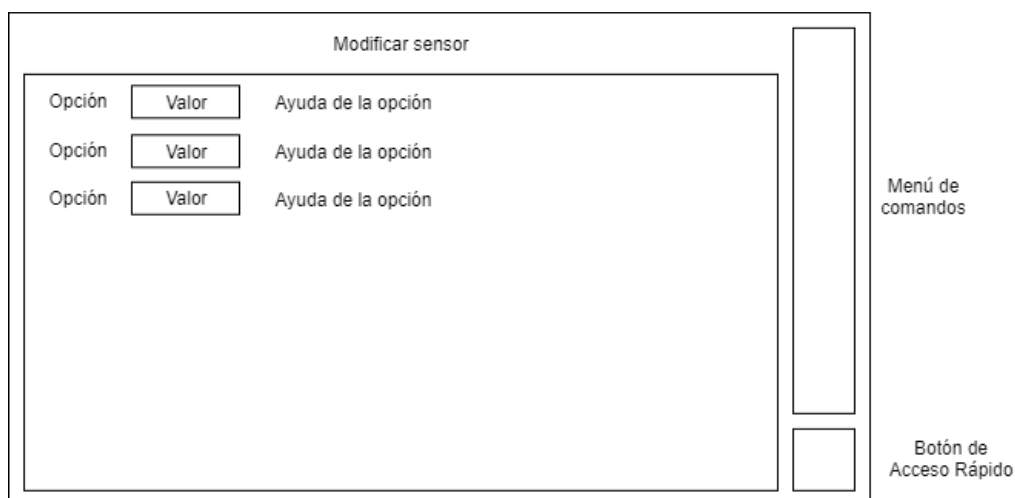
*Figura 4.8. Esquema de la pantalla de detalles de un sensor*

Esta pantalla muestra los detalles de un sensor, y dispone de las siguientes acciones:

- Volver atrás.

- Navegar a la pantalla de modificación para este sensor.
- Eliminar este sensor.

#### 4.6.1.6 Pantalla de modificación de un sensor

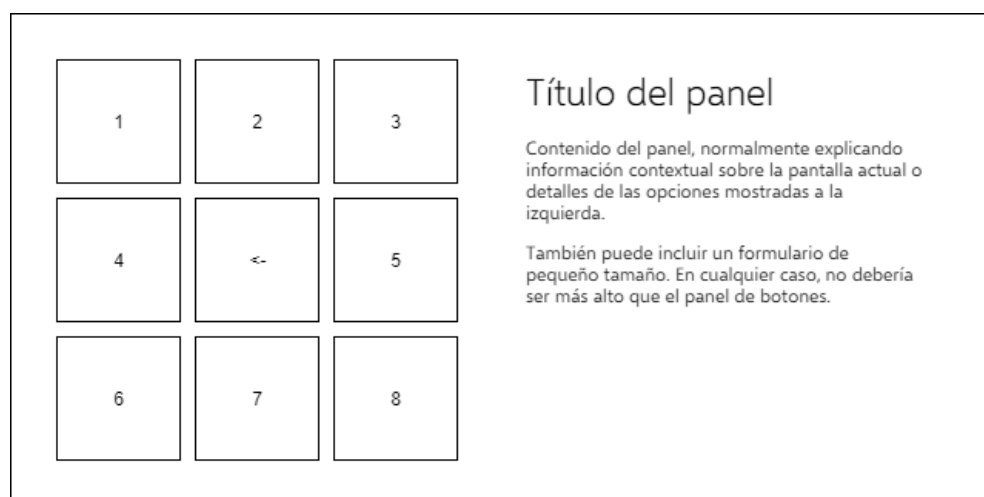


*Figura 4.9. Esquema de la pantalla de modificación de un sensor*

Esta pantalla permite modificar los atributos de un sensor, e incluye las siguientes acciones:

- Cancelar la modificación.
- Guardar los cambios.

#### 4.6.1.7 Interfaz del Acceso Rápido



*Figura 4.10. Esquema de la interfaz del Acceso Rápido*

Esta interfaz se muestra como un cuadro de diálogo centrado en la pantalla cuando el usuario accede al Acceso Rápido.

Tanto las opciones de la cuadrícula como el texto se modifican de forma contextual según el usuario navega por las opciones.

Cabe destacar que si no fuera necesario mostrar ocho opciones en un momento dado, los botones que sobren se harán invisibles.

## 4.6.2 Descripción del Comportamiento de la Interfaz

En general, la interfaz de usuario no modificará la distribución de la pantalla una vez que la navegación ha finalizado. Esto implica que, una vez que el usuario accede a una pantalla, puede tener la certeza de que todos los elementos que ve van a quedarse donde están, y ningún elemento nuevo va a aparecer.

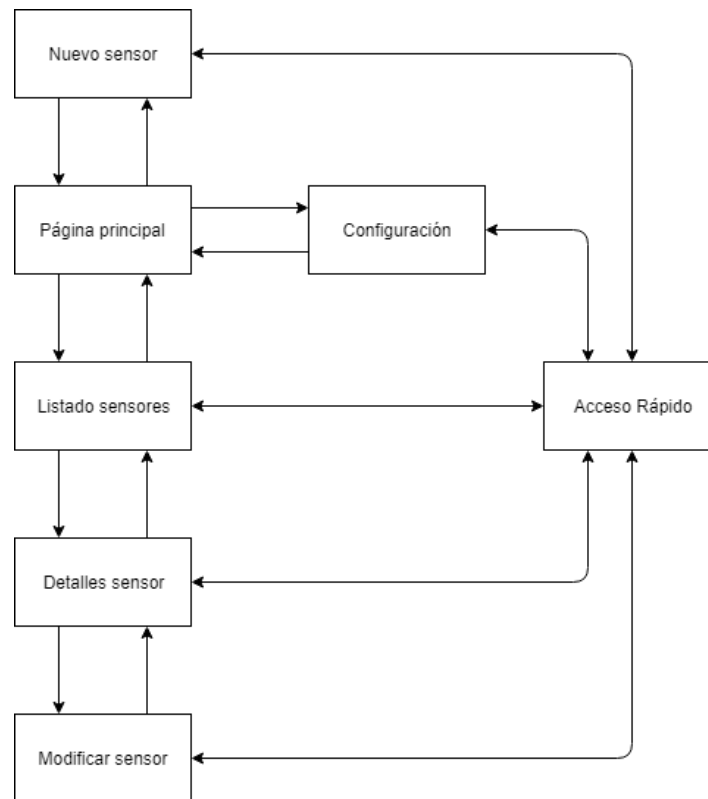
Como excepción a esta regla, en algunas pantallas es posible que un mensaje de notificación se deslice desde la parte inferior de la pantalla como respuesta a una acción directa del usuario. Este mensaje contendrá normalmente información contextual, y desaparecerá automáticamente al cabo de pocos segundos. Cabe destacar que la aparición de este mensaje no alterará el resto de elementos, sino que se colocará sobre ellos.

Cuando el usuario deba escribir datos, y estos no superen la validación, el sistema informará de esta circunstancia mediante un cuadro de diálogo que aparecerá en el centro de la pantalla. Este diálogo interrumpirá la interacción con el resto de elementos hasta que el usuario lo cierre.

De forma similar, los cuadros de diálogo se usarán cuando una página deba ofrecer al usuario la posibilidad de modificar su comportamiento en momentos puntuales (como por ejemplo la página de listados, que permite al usuario modificar el criterio de filtrado). Así se evita que los campos modificables estén siempre visibles ocupando espacio y distrayendo al usuario del propósito principal de la página.

### 4.6.3 Diagrama de Navegabilidad

En esta sección se incluye un diagrama que muestra la navegación entre las pantallas de la aplicación:



*Figura 4.11. Diagrama de navegabilidad*

## 4.7 Especificación del Plan de Pruebas

En esta sección se indican las pruebas que asegurarán el funcionamiento correcto del sistema.

### 4.7.1 Pruebas Unitarias

La ejecución de pruebas en el componente ConfigAdapter se ha automatizado mediante MSTest. Durante el desarrollo, estas pruebas se pueden ejecutar en cualquier momento. Por otro lado, al subir el código al sistema de control de versiones, un proceso de Integración Continua se encarga de pasar todas las pruebas para asegurarse de que el sistema sigue funcionando.

El resto de pruebas unitarias no está automatizado debido a la complejidad de funcionamiento de los componentes correspondientes. En este caso, las pruebas se ejecutan manualmente tras terminar el desarrollo de las funcionalidades clave, y todas juntas antes de la entrega final del sistema.

En cualquier caso, cuando se detecten fallos, se investigará su origen y se solucionará el problema lo antes posible, para evitar que los errores se acumulen.

Librería de configuración	
Prueba	Resultado Esperado
Intentar abrir un archivo con formato desconocido.	Se lanza una excepción.
Prueba	Resultado Esperado
Intentar abrir un archivo con formato incorrecto.	Se lanza una excepción.
Prueba	Resultado Esperado
Abrir un archivo con el formato correcto.	El archivo se abre correctamente.
Prueba	Resultado Esperado
Leer un ajuste global.	Se devuelve el valor del ajuste.
Prueba	Resultado Esperado
Leer un ajuste incluido en una categoría.	Se devuelve el valor del ajuste.
Prueba	Resultado Esperado
Intentar leer un ajuste con clave inválida.	Se lanza una excepción.
Prueba	Resultado Esperado
Leer un ajuste cuya clave no existe.	Se devuelve un valor nulo.
Prueba	Resultado Esperado
Escribir un ajuste global.	El ajuste se guarda correctamente.

Prueba	Resultado Esperado
Escribir un ajuste incluido en una categoría	El ajuste se guarda correctamente.
Prueba	Resultado Esperado
Modificar el valor de un ajuste.	El ajuste pasa a tener el valor modificado.
Prueba	Resultado Esperado
Intentar escribir un ajuste con clave inválida.	Se lanza una excepción.
Prueba	Resultado Esperado
Pedir todos los ajustes globales.	Se devuelven los ajustes globales.
Prueba	Resultado Esperado
Pedir todos los ajustes de una categoría.	Se devuelven los ajustes de esa categoría.
Prueba	Resultado Esperado
Eliminar un ajuste.	El ajuste se elimina del archivo.
Prueba	Resultado Esperado
Eliminar una categoría.	La categoría y todos sus ajustes se eliminan.

Aplicación local	
Prueba	Resultado Esperado
Conectarse a un servidor inexistente.	Se devuelve un cliente sin conexión.
Prueba	Resultado Esperado
Conectarse a un servidor correcto.	Se devuelve un cliente con conexión activa.
Prueba	Resultado Esperado
Comprobar la conexión de un cliente sin conexión.	Se devuelve un valor falso.
Prueba	Resultado Esperado
Comprobar la conexión de un cliente con conexión.	Se devuelve un valor verdadero.
Prueba	Resultado Esperado
Acoplarse a un proceso en un cliente sin conexión.	Se lanza una excepción.
Prueba	Resultado Esperado
Acoplarse a un proceso inexistente en un cliente con conexión.	Se lanza una excepción.
Prueba	Resultado Esperado
Acoplarse a un proceso existente en un cliente con conexión.	El cliente se acopla al proceso indicado.
Prueba	Resultado Esperado
Comprobar si hay	Se devuelve un valor nulo.

resultados en un cliente sin conexión.	
Prueba	Resultado Esperado
Comprobar si hay resultados en un cliente con conexión y un proceso sin resultados.	Se devuelve un valor falso.
Prueba	Resultado Esperado
Comprobar si hay resultados en un cliente con conexión y un proceso con resultados.	Se devuelve un valor verdadero.
Prueba	Resultado Esperado
Conseguir los resultados de un cliente sin conexión.	Se devuelve un valor nulo.
Prueba	Resultado Esperado
Conseguir los resultados de un cliente con conexión y un proceso sin resultados.	Se devuelve un valor nulo.
Prueba	Resultado Esperado
Conseguir los resultados de un cliente con conexión y un proceso con resultados.	Se devuelven los resultados más recientes del proceso.
Prueba	Resultado Esperado
Validar una dirección IP correcta.	Se devuelve un valor verdadero.
Prueba	Resultado Esperado
Validar una dirección IP incorrecta.	Se devuelve un valor falso.
Prueba	Resultado Esperado
Validar un puerto correcto.	Se devuelve un valor verdadero.
Prueba	Resultado Esperado
Validar un puerto incorrecto	Se devuelve un valor falso.

Servidor remoto	
Prueba	Resultado Esperado
Descargar un archivo de una URI inexistente.	Se lanza una excepción.
Prueba	Resultado Esperado
Descargar un archivo de una URI existente.	El archivo se descarga correctamente.
Prueba	Resultado Esperado
Descargar un archivo sobrescribiendo el destino.	Se sobrescribe el archivo local con el archivo descargado.
Prueba	Resultado Esperado
Cargar un ensamblado	Se lanza una excepción.



inexistente.	
Prueba	Resultado Esperado
Cargar un ensamblado incompatible.	Se lanza una excepción.
Prueba	Resultado Esperado
Cargar un ensamblado al que le faltan referencias.	Se lanza una excepción.
Prueba	Resultado Esperado
Cargar un ensamblado correcto.	El ensamblado se carga en memoria.
Prueba	Resultado Esperado
Listar los ensamblados cargados.	Se devuelve una lista de ensamblados cargados.
Prueba	Resultado Esperado
Listar los ensamblados cuando no hay ninguno cargado.	Se devuelve una lista vacía.
Prueba	Resultado Esperado
Descargar un ensamblado que no está cargado.	El sistema no hace nada.
Prueba	Resultado Esperado
Descargar un ensamblado cargado.	El ensamblado se descarga de la memoria.
Prueba	Resultado Esperado
Instanciar un proceso de un ensamblado que no está cargado.	El sistema no hace nada.
Prueba	Resultado Esperado
Instanciar un proceso inexistente.	Se lanza una excepción.
Prueba	Resultado Esperado
Instanciar un proceso que ya está en ejecución.	Se lanza una excepción.
Prueba	Resultado Esperado
Instanciar un proceso que no está en ejecución.	El proceso se instancia correctamente.
Prueba	Resultado Esperado
Listar los procesos en ejecución.	Se devuelve una lista de procesos en ejecución.
Prueba	Resultado Esperado
Listar los procesos cuando ninguno está en ejecución.	Se devuelve una lista vacía.

## 4.7.2 Pruebas de Integración y del Sistema

Las pruebas de integración y del sistema se llevarán a cabo tras implementar la funcionalidad correspondiente a cada una.

Además, se probarán todas juntas antes de terminar el desarrollo para garantizar que el sistema se comporta de la manera esperada.

Gestionar sensores: Ver distribución de sensores	
Prueba	Resultado Esperado
Abrir la aplicación sin tener ningún sensor conectado.	El sistema muestra el mapa sin ningún icono.
Prueba	Resultado Esperado
Abrir la aplicación teniendo dos sensores en países diferentes.	El sistema muestra el mapa con dos iconos, uno en cada país.
Prueba	Resultado Esperado
Abrir la aplicación teniendo dos sensores en el mismo país.	El sistema muestra el mapa con un solo icono.

Gestionar sensores: Ver listado de sensores	
Prueba	Resultado Esperado
Ir al listado sin tener ningún sensor conectado.	El sistema muestra un listado vacío.
Prueba	Resultado Esperado
Ir al listado de todos los sensores.	El sistema muestra el listado con todos los sensores conectados.
Prueba	Resultado Esperado
Ir al listado desde uno de los iconos de mapa.	El sistema muestra el listado con los sensores de ese país.

Gestionar sensores: Filtrar listado de sensores	
Prueba	Resultado Esperado
Filtrar un listado vacío.	El sistema muestra un listado vacío.
Prueba	Resultado Esperado
Aplicar un criterio cumplido por todos los sensores.	El sistema muestra el listado con todos los sensores.
Prueba	Resultado Esperado
Aplicar un criterio cumplido por alguno de los sensores.	El sistema muestra el listado con los sensores que cumplen el criterio.
Prueba	Resultado Esperado
Aplicar un criterio no cumplido por ningún sensor.	El sistema muestra un listado vacío.

Gestionar sensores: Modificar filtros de sensores	
Prueba	Resultado Esperado
Aplicar la modificación sin hacer ningún cambio.	El sistema muestra el mismo listado que antes.
Prueba	Resultado Esperado
Aplicar la modificación habiendo hecho cambios.	El sistema muestra un listado actualizado según los nuevos criterios.
Prueba	Resultado Esperado
Eliminar todos los filtros activos.	El sistema muestra un listado con todos los sensores conectados.

Gestionar sensores: Añadir un nuevo sensor	
Prueba	Resultado Esperado
Añadir un sensor de forma correcta.	El sistema tiene un sensor más que antes.
Prueba	Resultado Esperado
Introducir mal los valores de conexión.	El sistema muestra un error y no permite añadir el sensor.
Prueba	Resultado Esperado
Cancelar la operación antes de añadir un sensor.	El sistema no cambia el número de sensores conectados.

Gestionar sensores: Modificar un sensor existente	
Prueba	Resultado Esperado
Cambiar los valores de un sensor.	El sistema muestra los nuevos valores al ver los detalles de ese sensor.
Prueba	Resultado Esperado
Guardar los cambios manteniendo los valores anteriores.	El sistema muestra los valores originales del sensor.
Prueba	Resultado Esperado
Cancelar la modificación.	El sistema muestra los valores originales del sensor.

Gestionar sensores: Eliminar un sensor	
Prueba	Resultado Esperado
Aceptar la confirmación.	El sistema muestra un sensor menos que antes.
Prueba	Resultado Esperado
Denegar la confirmación.	El sistema muestra los mismos sensores que antes.

Ver estado de un sensor	
Prueba	Resultado Esperado
Ver el estado de un sensor desconectado.	El sistema muestra los parámetros del sensor, pero no sus propiedades.
Prueba	Resultado Esperado
Ver el estado de un sensor conectado.	El sistema muestra el listado con todos los sensores conectados.

Modificar la configuración del sistema	
Prueba	Resultado Esperado
Aplicar la modificación sin hacer ningún cambio.	El sistema se reinicia y muestra los mismos valores de configuración que antes.
Prueba	Resultado Esperado
Aplicar la modificación habiendo hecho cambios.	El sistema se reinicia y muestra los nuevos valores de configuración.
Prueba	Resultado Esperado
Cancelar la modificación.	El sistema no se reinicia y mantiene los mismos valores de configuración.

### 4.7.3 Pruebas de Usabilidad

Las pruebas de usabilidad se utilizan para comprobar si los usuarios finales son capaces de utilizar el sistema de forma eficiente sin dificultades excesivas.

En este caso, se plantea el siguiente desarrollo para estas pruebas:

- **Usuarios:** Las pruebas se ejecutarán por varios empleados del cliente que tengan o puedan llegar a tener alguna interacción con el sistema una vez terminado.
- **Lugar de realización:** En las oficinas del cliente.
- **Metodología:** Se ofrecerá un cuestionario a los usuarios para indicar las tareas que deben realizar, y averiguar cómo es su experiencia durante el proceso.

### 4.7.3.1 Cuestionario de Evaluación

En primer lugar, se harán algunas preguntas de carácter general para hacerse una idea del perfil de usuario:

<p>¿Usa un ordenador frecuentemente?</p> <ol style="list-style-type: none"> <li>1. Todos los días</li> <li>2. Varias veces a la semana</li> <li>3. Ocasionalmente</li> <li>4. Nunca o casi nunca</li> </ol>
<p>¿Qué tipo de actividades realiza con el ordenador?</p> <ol style="list-style-type: none"> <li>1. Es parte de mi trabajo o profesión</li> <li>2. Lo uso básicamente para ocio</li> <li>3. Solo empleo aplicaciones estilo Office</li> <li>4. Únicamente leo el correo y navego ocasionalmente</li> </ol>
<p>¿Ha usado alguna vez software como el de esta prueba?</p> <ol style="list-style-type: none"> <li>1. Sí, he empleado software similar</li> <li>2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares</li> <li>3. No, nunca</li> </ol>
<p>¿Qué busca principalmente en un programa?</p> <ol style="list-style-type: none"> <li>1. Que sea fácil de usar</li> <li>2. Que sea intuitivo</li> <li>3. Que sea rápido</li> <li>4. Que tenga todas las funciones necesarias</li> </ol>

A continuación, se propondrán las siguientes actividades, para que los usuarios las hagan e informen de posibles dificultades durante el proceso:

- Importar un fichero de configuración.
- Acceder al listado de sensores y cambiar el criterio de filtrado.
- Conectar un nuevo sensor al sistema.
- Ver los detalles del sensor que se acaba de conectar.
- Modificar alguno de los datos del sensor que se acaba de conectar.

Por último, el usuario responderá a algunas preguntas específicas:

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Comprende la estructura de navegación de la aplicación?</i>				
<i>¿En algún momento siente que debería recibir ayuda?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿En algún momento el sistema actúa de forma inesperada?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>Los textos de la interfaz son...</i>				
<i>Los iconos e imágenes son...</i>				
<i>Los colores empleados son...</i>				
Diseño de la Interfaz	Si		No	A veces
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
Observaciones				

#### 4.7.3.2 Cuestionario para el Responsable de las Pruebas

Además, durante las pruebas, el responsable irá rellenando su propio cuestionario en función del comportamiento que observa en los usuarios:

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	
<i>El usuario no necesita demasiado tiempo para cada tarea</i>	
<i>El usuario comete errores leves</i>	
<i>El usuario comete errores graves</i>	

<i>El usuario navega de forma innecesaria</i>	
<i>El usuario parece desorientado en cuanto a la estructura de la aplicación</i>	

## 4.7.4 Pruebas de Rendimiento

Para las pruebas de rendimiento se han considerado ordenadores con características similares a los que el cliente empleará para poner en ejecución la aplicación:

- Procesador Intel Core i7 a 2,8 GHz.
- 16 GB de memoria RAM.
- Disco duro SSD.

Además, se harán las pruebas con una cantidad de sensores estimada como máximo teórico (en base a las estimaciones de carga de trabajo ofrecidas por el cliente) de 30 sensores.

Durante las pruebas, se usarán dos equipos:

- Uno que ejecutará la aplicación local.
- Otro que tendrá los servidores remotos.

Estos dos equipos estarán dentro de la misma red local.

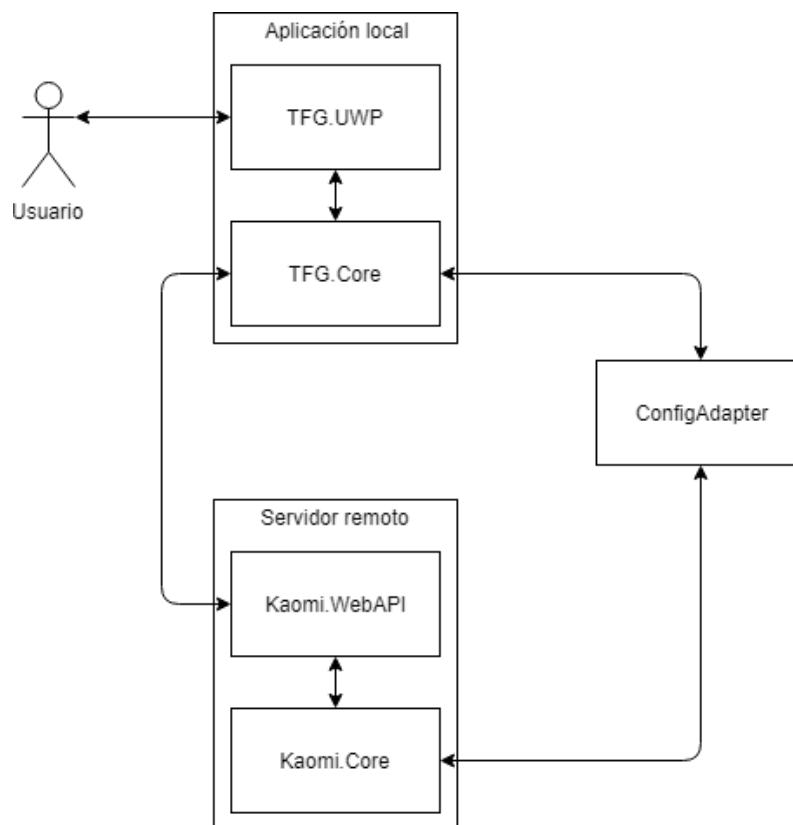
Las pruebas de rendimiento buscan asegurarse de que la aplicación local es capaz de responder con agilidad a los comandos del usuario en una situación similar al uso real.

## Capítulo 5. Diseño del Sistema

### 5.1 Arquitectura del Sistema

#### 5.1.1 Diagramas de Paquetes

Este es el diagrama de paquetes del proyecto. Cada uno de esos paquetes será explicado en detalle más adelante:

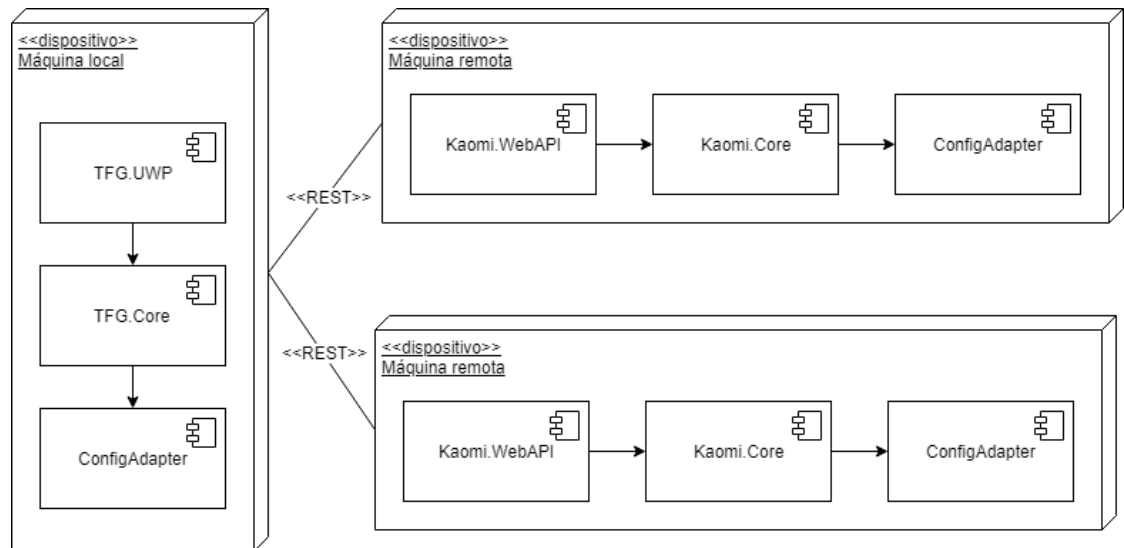


*Figura 5.1. Diagrama de paquetes del proyecto*



## 5.1.2 Diagramas de Despliegue

Este diagrama muestra cómo se distribuyen los componentes del sistema una vez desplegado:



*Figura 5.2. Diagrama de despliegue del sistema*

### 5.1.2.1 Máquina local

Representa el dispositivo en el que el usuario ha instalado la aplicación.

El componente TFG.Core actuará como cliente web, conectándose a las máquinas remotas para conseguir datos de sus sensores instalados.

### 5.1.2.2 Máquina remota

Representa cada uno de los dispositivos que tienen sensores conectados.

El componente Kaomi.WebAPI actuará como servidor web para devolver datos de su sensor cuando la interfaz de usuario se lo solicite.

## 5.2 Diseño de Clases

### 5.2.1 TFG.UWP

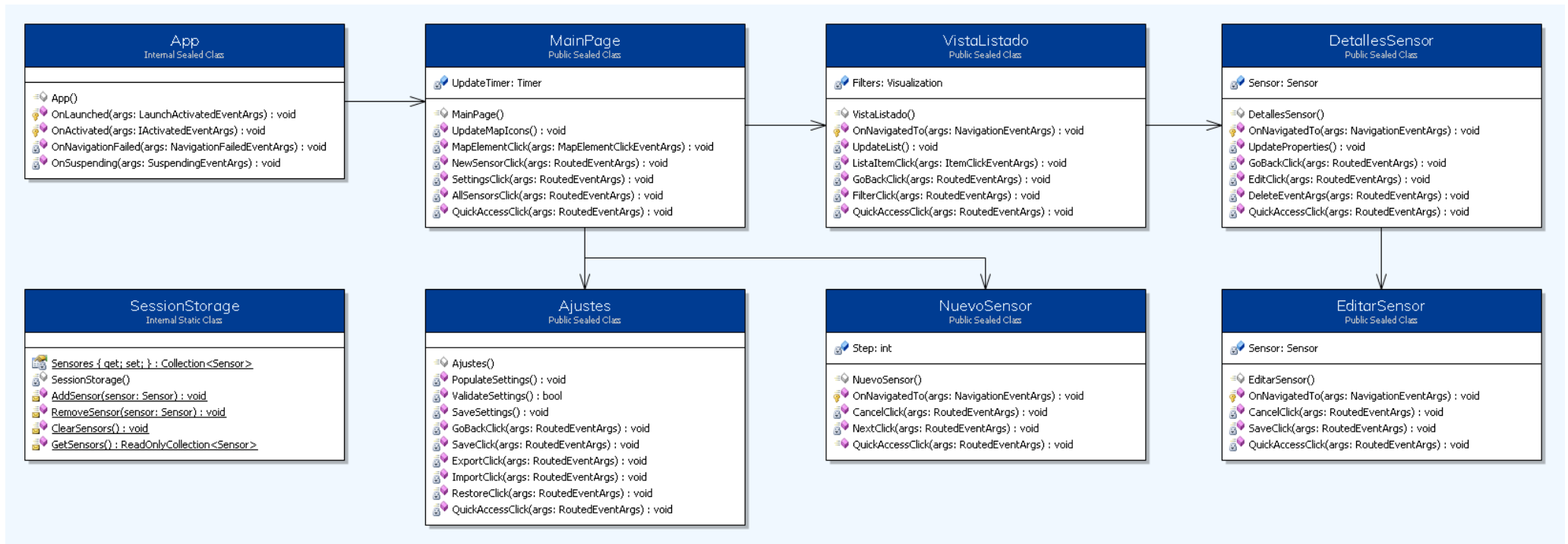


Figura 5.3. Diagrama de clases principal del paquete TFG.UWP

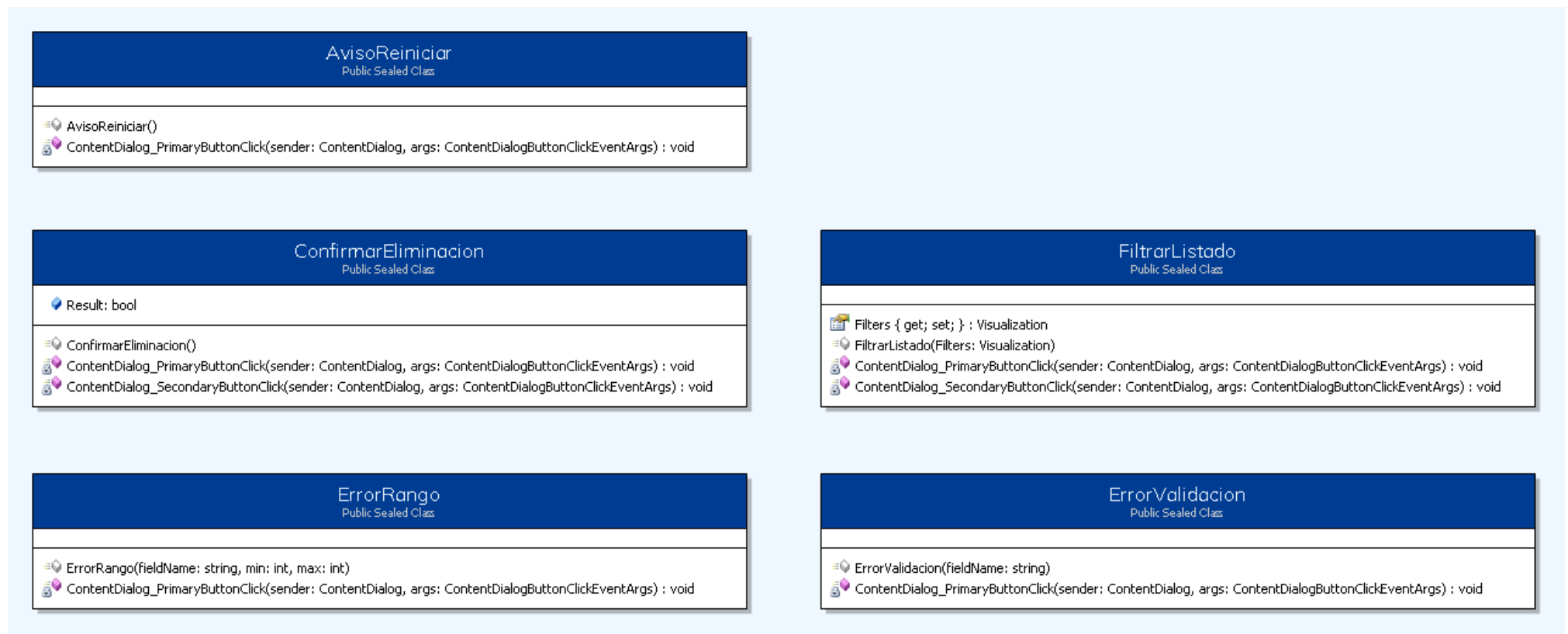


Figura 5.4. Diagrama de clases de diálogos del paquete TFG.UWP

## 5.2.2 TFG.Core

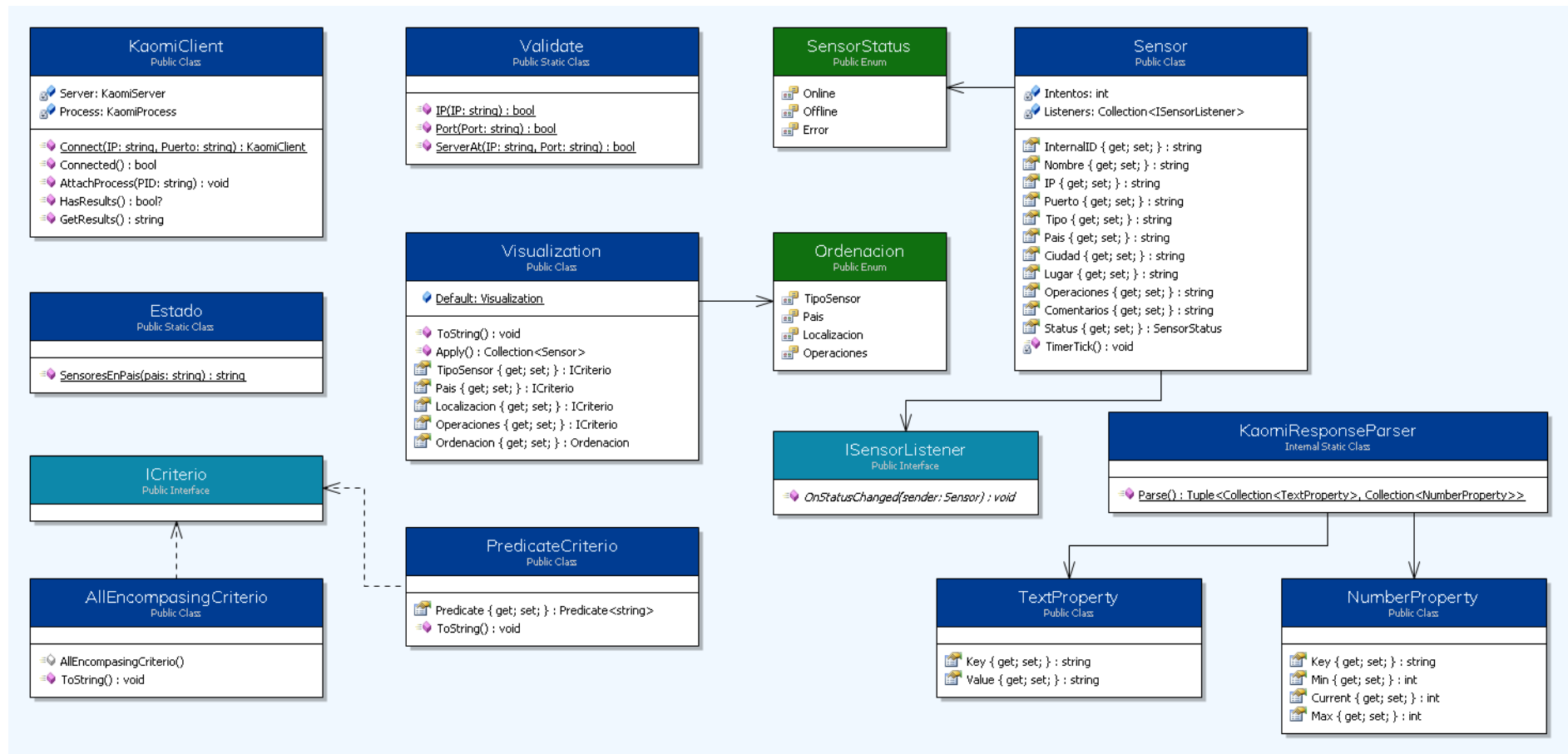


Figura 5.5. Diagrama de clases principal del paquete TFG.Core

### 5.2.3 Kaomi.Core

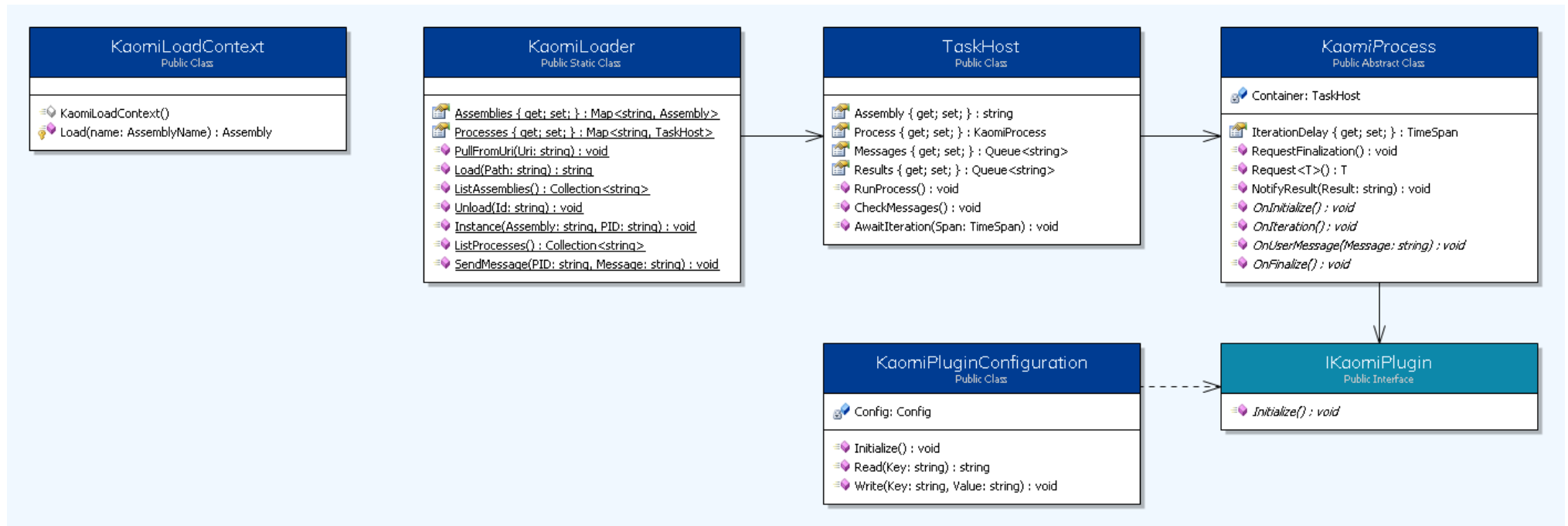
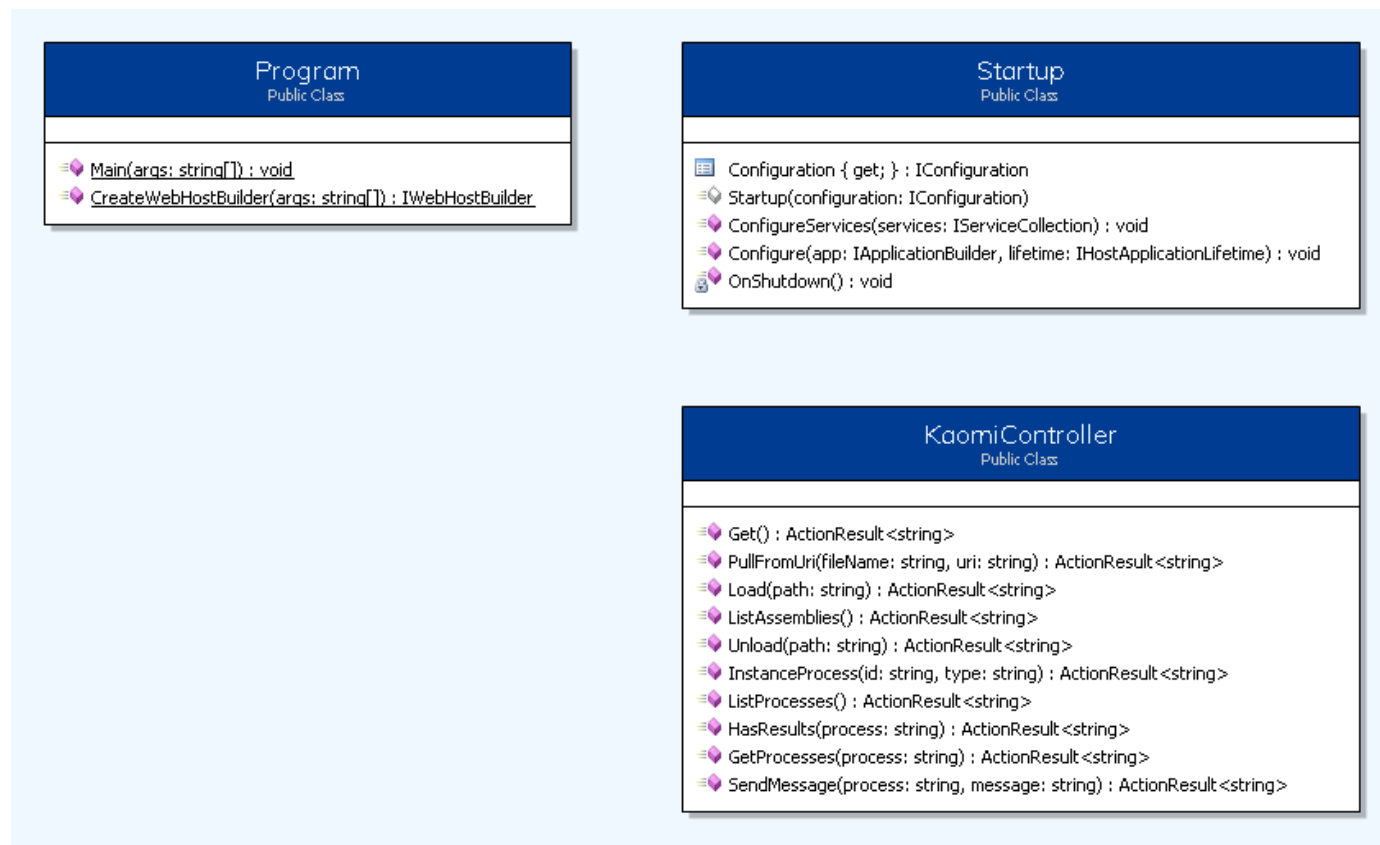


Figura 5.6. Diagrama de clases principal del paquete Kaomi.Core

## 5.2.4 Kaomi.WebAPI



*Figura 5.7. Diagrama de clases principal del paquete Kaomi.WebAPI*

## 5.2.5 ConfigAdapter

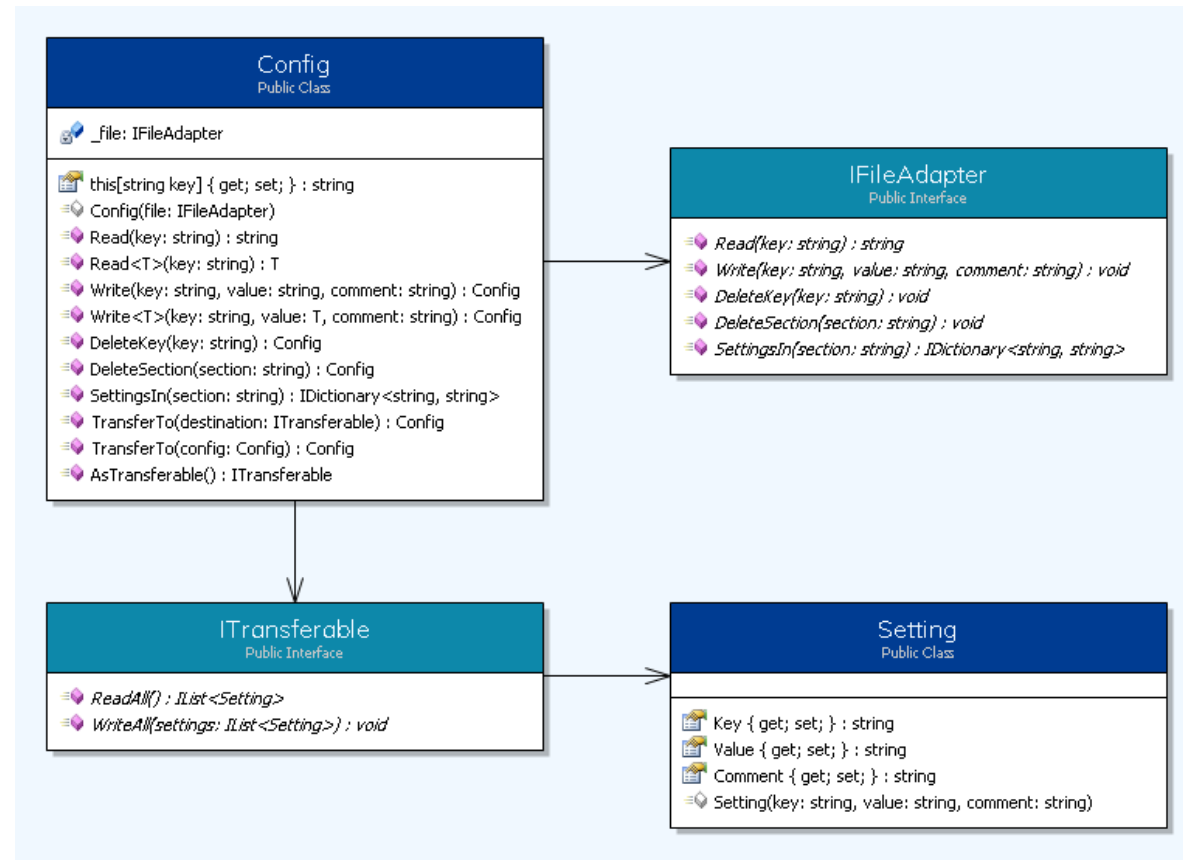


Figura 5.8. Diagrama de clases principal del paquete ConfigAdapter

## 5.2.6 ConfigAdapter.Xml

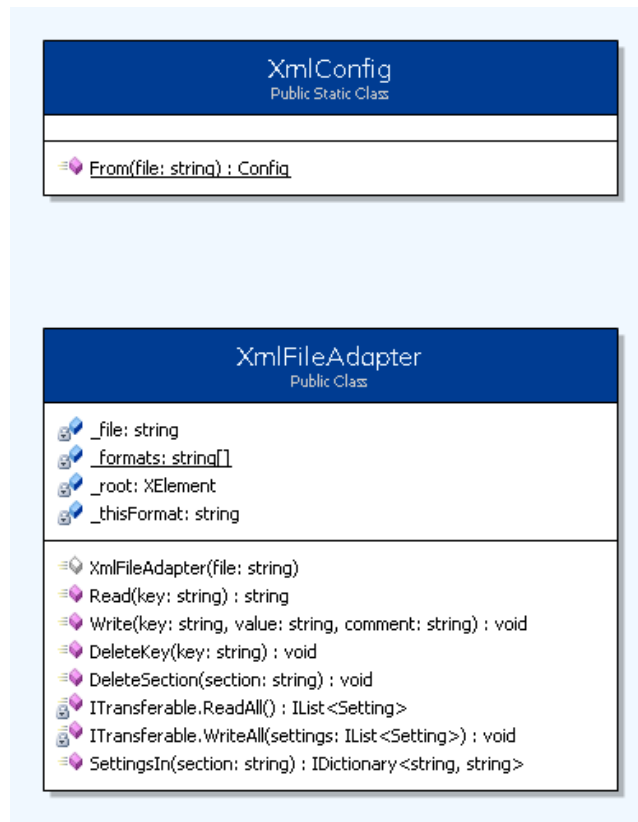
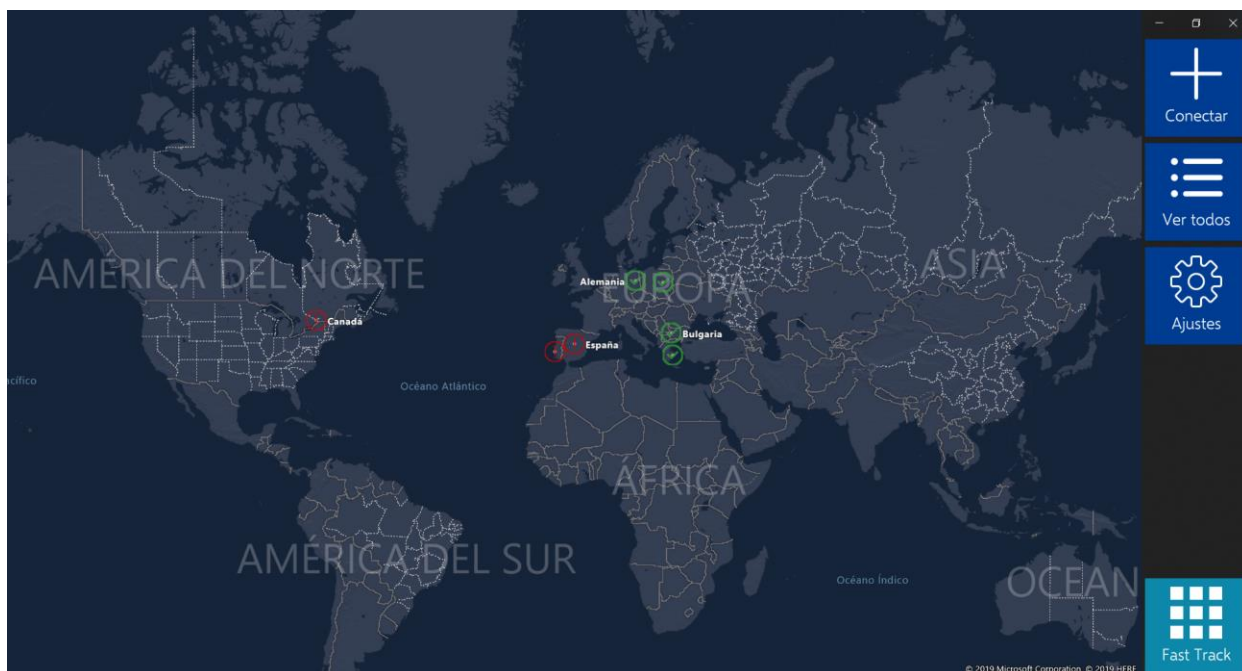


Figura 5.9. Diagrama de clases principal del paquete `ConfigAdapter.Xml`

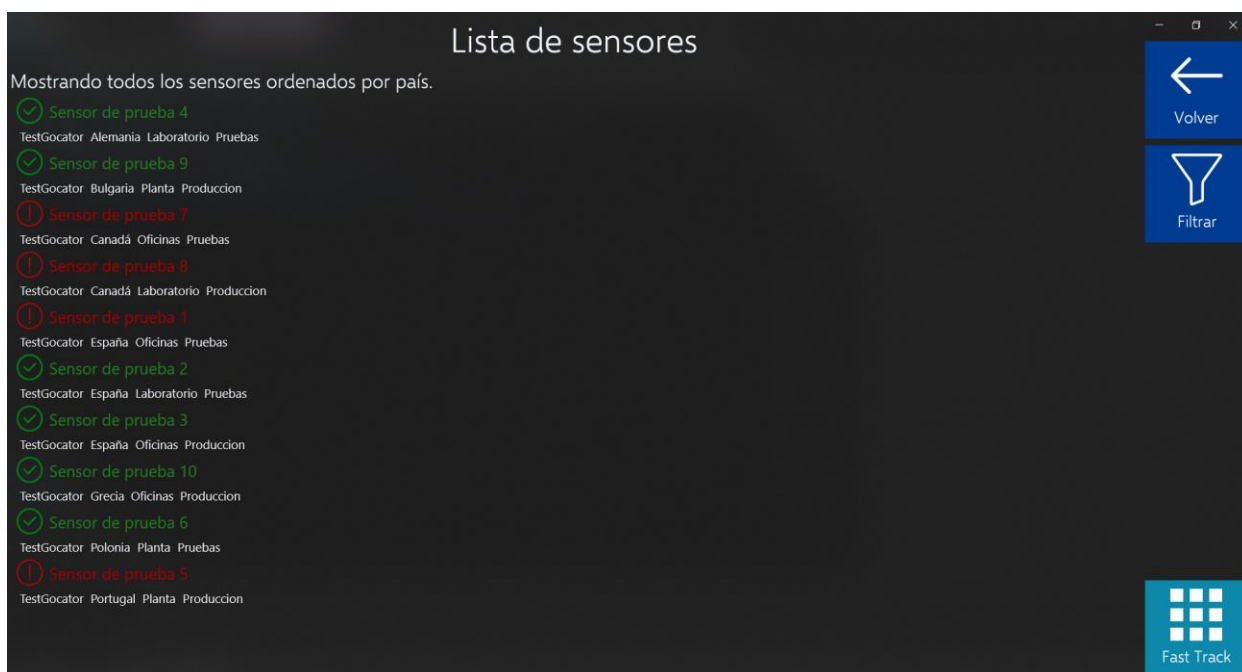


## 5.3 Diseño de la Interfaz

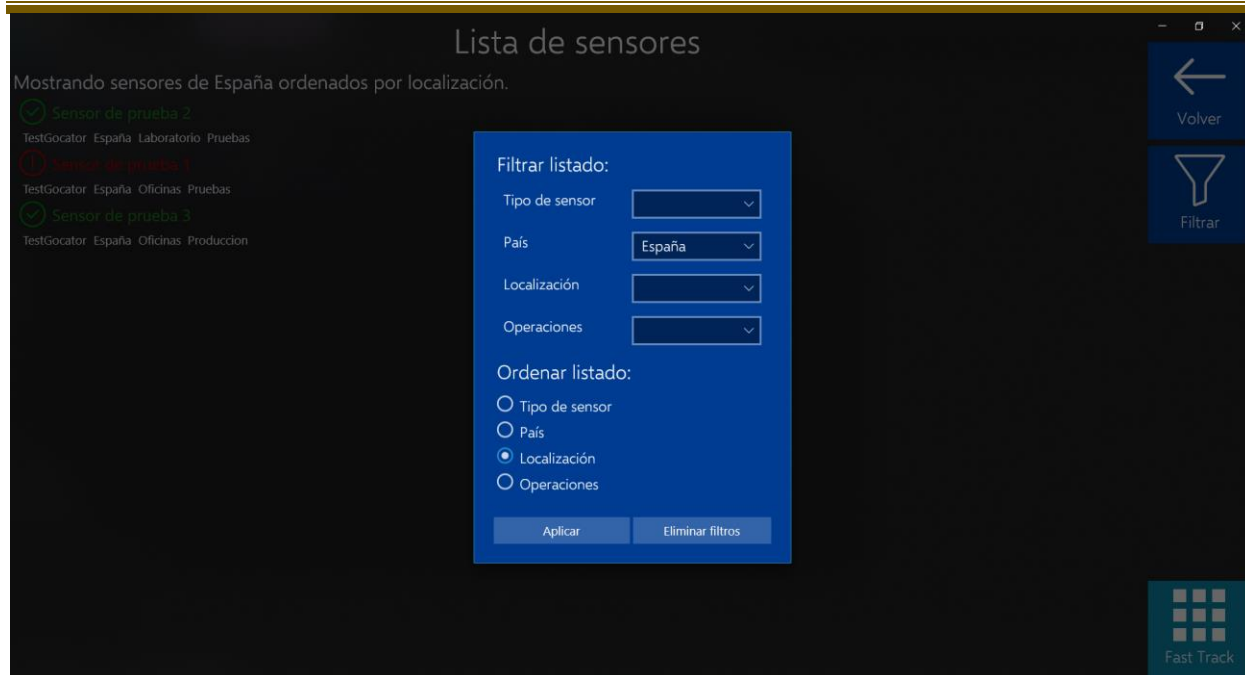
En esta sección se mostrarán los diseños definitivos de la interfaz de usuario, basados en los diseños preliminares que se pueden encontrar en la sección de análisis correspondiente.



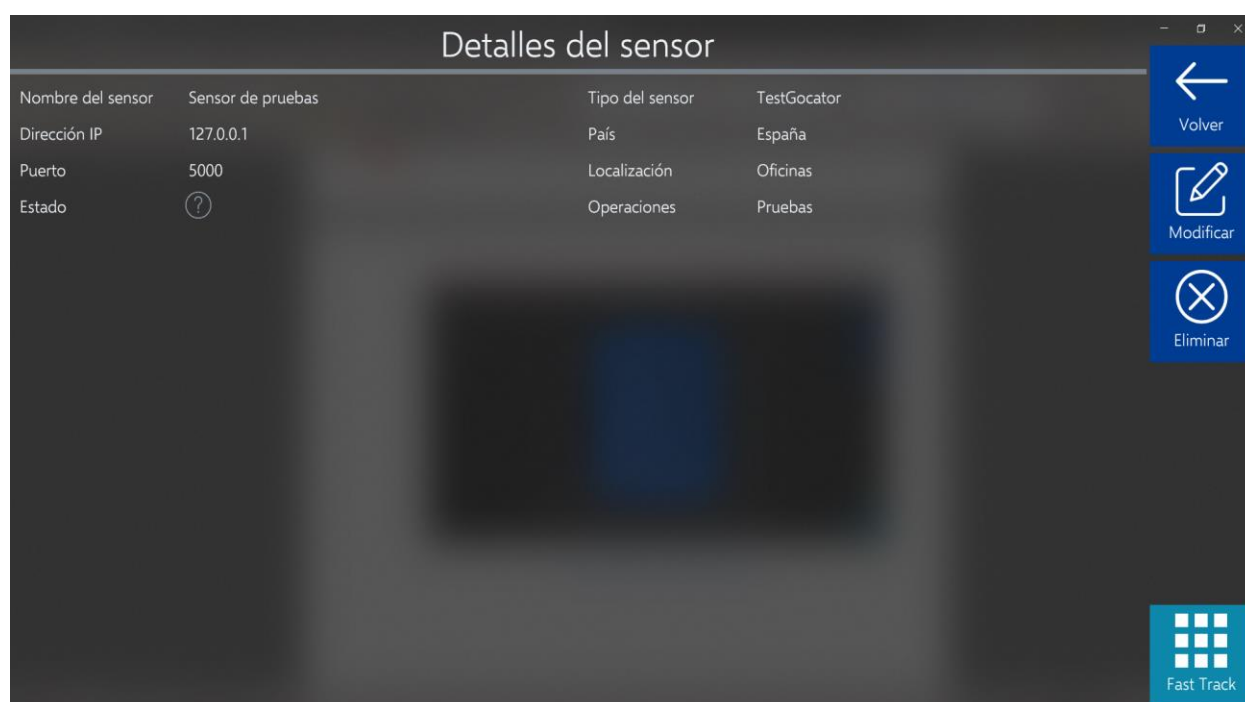
*Figura 5.10: Pantalla inicial de la aplicación.*



*Figura 5.11: Pantalla de listado de sensores.*



*Figura 5.12: Diálogo para filtrar el listado.*



*Figura 5.13: Pantalla de detalles de un sensor.*

**Modificar sensor**

Nombre:  Nombre que se mostrará asociado al sensor.

Dirección IP:  Punto de conexión al servidor remoto donde está alojado el sensor.

Puerto:  Punto de conexión al servidor remoto donde está alojado el sensor.

Tipo de sensor:  Indica de qué tipo es el sensor (y qué proceso modular debe usarse).

País:  Usado con fines de filtrado y clasificación en la interfaz.

Localización:  Usado con fines de filtrado y clasificación en la interfaz.

Ciudad:  Usado únicamente para facilitar la identificación.

Operaciones:  Usado con fines de filtrado y clasificación en la interfaz.

Comentarios:  Texto arbitrario para añadir cualquier otro detalle útil.

Cancelar

Guardar

Fast Track

*Figura 5.14: Pantalla de modificación de un sensor.*

**Conectar nuevo sensor**

Vamos a conectar un nuevo sensor al sistema. Antes de empezar, asegúrate de que el servidor remoto está correctamente instalado y con el proceso adecuado en ejecución. Además, tanto este equipo como el equipo remoto deben tener conexión a internet.

Para empezar, introduce los siguientes datos básicos sobre el servidor remoto:

Dirección IP:

Puerto:

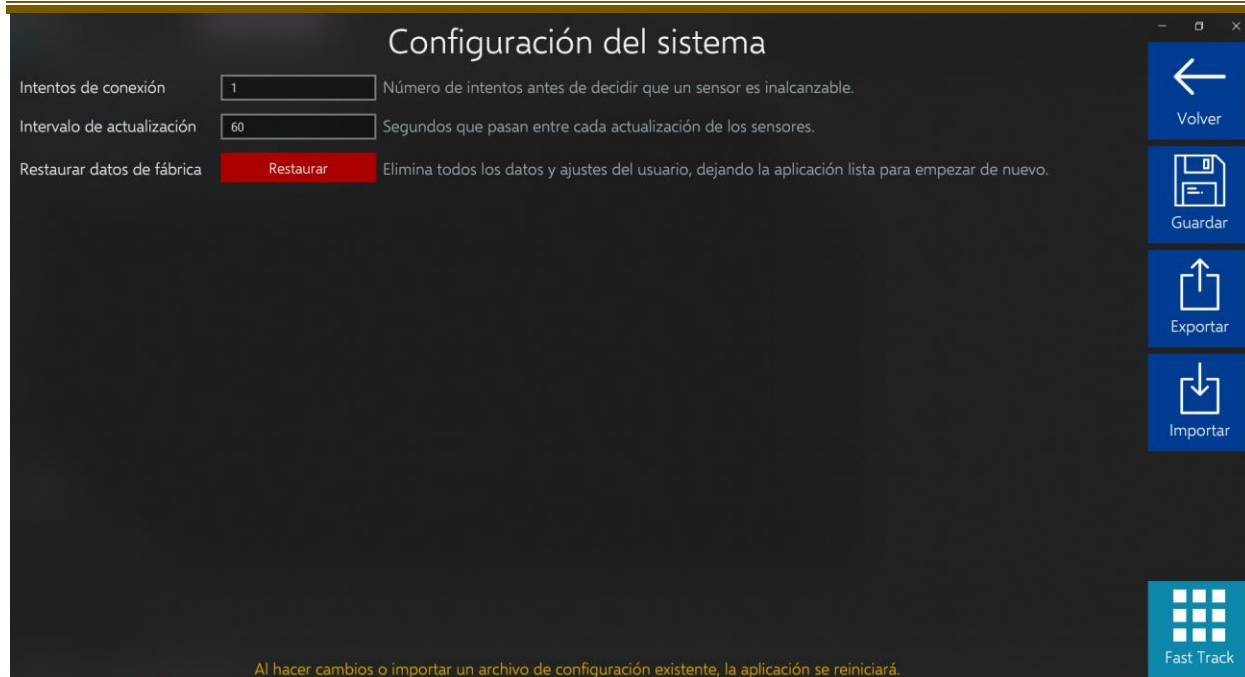
Tipo de sensor:

Cancelar

Continuar

Fast Track

*Figura 5.15: Pantalla de conexión de un sensor nuevo.*



*Figura 5.16: Pantalla de ajustes del sistema.*



*Figura 5.17: Panel de Acceso Rápido.*

# Capítulo 6. Implementación del Sistema

## 6.1 Estándares y Normas Seguidos

El diseño del componente TFG.UWP se ha hecho de acuerdo con el estándar Fluent Design System de Microsoft.

## 6.2 Lenguajes de Programación

El lenguaje de programación utilizado durante el desarrollo ha sido C#, en su versión 8.0, tal como se distribuye en la versión 16.4 de Visual Studio.

Los siguientes componentes forman parte relevante de la instalación de Visual Studio:

- Administrador de paquetes NuGet 5.3.0
- Herramientas de C# 3.3.1
- MSBuild 15.0

Las siguientes librerías y ensamblados forman parte del proyecto (ya sea directamente o como dependencia interna de alguna librería o ensamblado usado directamente):

- DotNet.Misc.Extensions 1.2.1
- Microsoft.AspNetCore 2.2.0
- Microsoft.AspNetCore.Mvc 2.2.0
- Microsoft.Net.Native.Compiler 2.2.7
- Microsoft.Net.UWPCoreRuntimeSdk 2.2.9
- Microsoft.NETCore.Platforms 1.1.0
- Microsoft.NETCore.Targets 1.1.0
- Microsoft.NETCore.UniversalWindowsPlatform 6.2.9
- Microsoft.Toolkit.UWP.Notifications 5.1.1
- Microsoft.Toolkit.UWP.UI.Controls 5.1.1
- NetStandard.Library 2.0.3
- Newtonsoft.Json 12.0.2
- QueryString.NET 1.0.0
- System.Collections 4.3.0
- System.Diagnostics.Debug 4.3.0
- System.Globalization 4.3.0
- System.IO 4.3.0
- System.ObjectModel 4.3.0
- System.Reflection 4.3.0
- System.Reflection.Primitives 4.3.0
- System.Resources.ResourceManager 4.3.0
- System.Runtime 4.3.1
- System.Runtime.Loader 4.3.0
- System.Runtime.WindowsRuntime 4.3.0
- System.Text.Encoding 4.3.0
- System.Threading 4.3.0
- System.Threading.Tasks 4.3.0
- Windows.Foundation.UniversalAPIContract 2.0.0

## 6.3 Herramientas y Programas Usados para el Desarrollo

En esta sección se indican las principales herramientas y entornos usados durante el desarrollo del sistema.

### 6.3.1 C#

Lenguaje de programación usado en todas las partes y componentes del sistema. La versión empleada es la 8.0

### 6.3.2 Visual Studio

Entorno de Desarrollo Integrado usado para escribir y depurar el código del proyecto. La versión empleada es la 16.4

### 6.3.3 Visual Studio Code

Editor de código ligero usado principalmente para modificar archivos de configuración e interactuar con el control de versiones. La versión empleada es la 1.38

### 6.3.4 Framework .NET

Sistema de compilación en tiempo real sobre el que se ejecuta el sistema. La versión empleada es la 4.6.1

### 6.3.5 .NET Core

Implementación moderna multiplataforma del framework .NET diseñada para tener una API uniforme sobre cualquier sistema operativo. La versión empleada es la 3.0

### 6.3.6 ASP.NET Core

Framework sucesor de ASP.NET, diseñado para crear aplicaciones web o servicios de WebAPI basados en .NET Core. La versión empleada es la 3.0

## 6.3.7 Plataforma UWP

Conjunto de librerías y entornos de ejecución que permiten aislar las aplicaciones en un contenedor para facilitar la instalación, actualización y desinstalación. La versión empleada es la 10.0.17134.12, correspondiente a la versión 1803 de Windows 10.

## 6.3.8 Azure DevOps

Sistema de control de versiones que permite gestionar el código y asociarle procesos de integración continua. Al ser un sistema basado en la web, no hay un número de versión disponible públicamente.

## 6.3.9 NuGet

Plataforma de distribución de paquetes para proyectos .NET que simplifica la gestión de las dependencias. Al ser un sistema basado en la web, no hay un número de versión disponible públicamente.

## 6.3.10 Mapas de Bing

API que permite a las aplicaciones UWP interactuar con mapas, incluyendo funciones como marcado de puntos, cálculo de rutas o geovallas. Esta API está asociada con la plataforma UWP y comparte su versión.

## 6.3.11 Peticiones REST

Protocolo de comunicaciones sin estado basado en mensajes de texto sobre HTTP. Este sistema no implementa una especificación concreta, por lo que no existe una versión específica.



## 6.4 Creación del Sistema

Esta sección incluye detalles adicionales sobre el proceso de desarrollo del sistema.

### 6.4.1 Problemas Encontrados

#### 6.4.1.1 Framework de las librerías de sensores

Las librerías que ofrece el fabricante de los sensores son anteriores a .NET Standard, y usan componentes específicos de Windows como COM. Por lo tanto, intentar utilizarlos desde una aplicación compilada para netcoreapp3.0 resulta en un error de ejecución.

Para solucionarlo, es necesario hacer que el componente Kaomi (que es el que las va a usar) esté compilado para netcoreapp2.0, de forma que sea compatible con net461.

#### 6.4.1.2 API de AssemblyLoadContext

Como resultado de solucionar el problema anterior, se descubre que la API encargada de descargar ensamblados de la memoria no está disponible cuando se compila para netcoreapp2.0, ya que fue añadida posteriormente. Esto implica que el componente Kaomi podría, en circunstancias muy poco probables, crear una fuga de memoria.

Esto se ha solucionado haciendo que el componente Kaomi se compile para net461, y que use la antigua API AppDomain, predecesora de AssemblyLoadContext.



## Capítulo 7. Desarrollo de las Pruebas

### 7.1 Pruebas Unitarias

Estos resultados corresponden a la ejecución final de las pruebas unitarias antes de la entrega del sistema.

ConfigAdapter	
Prueba	Resultado Esperado
Intentar abrir un archivo con formato desconocido.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Intentar abrir un archivo con formato incorrecto.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Abrir un archivo con el formato correcto.	El archivo se abre correctamente.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Leer un ajuste global.	Se devuelve el valor del ajuste.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Leer un ajuste incluido en una categoría.	Se devuelve el valor del ajuste.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Intentar leer un ajuste con clave inválida.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Leer un ajuste cuya clave no existe.	Se devuelve un valor nulo.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado

Escribir un ajuste global.	El ajuste se guarda correctamente.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Escribir un ajuste incluido en una categoría	El ajuste se guarda correctamente.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Modificar el valor de un ajuste.	El ajuste pasa a tener el valor modificado.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Intentar escribir un ajuste con clave inválida.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Pedir todos los ajustes globales.	Se devuelven los ajustes globales.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Pedir todos los ajustes de una categoría.	Se devuelven los ajustes de esa categoría.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Eliminar un ajuste.	El ajuste se elimina del archivo.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Eliminar una categoría.	La categoría y todos sus ajustes se eliminan.
	<b>Resultado obtenido</b>
	Correcto

<b>TFG.Core</b>	
<b>Prueba</b>	<b>Resultado Esperado</b>
Conectarse a un servidor inexistente.	Se devuelve un cliente sin conexión.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Conectarse a un servidor	Se devuelve un cliente con conexión activa.

correcto.	
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Comprobar la conexión de un cliente sin conexión.	Se devuelve un valor falso.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Comprobar la conexión de un cliente con conexión.	Se devuelve un valor verdadero.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Acoplarse a un proceso en un cliente sin conexión.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Acoplarse a un proceso inexistente en un cliente con conexión.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Acoplarse a un proceso existente en un cliente con conexión.	El cliente se acopla al proceso indicado.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Comprobar si hay resultados en un cliente sin conexión.	Se devuelve un valor nulo.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Comprobar si hay resultados en un cliente con conexión y un proceso sin resultados.	Se devuelve un valor falso.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Comprobar si hay resultados en un cliente con	Se devuelve un valor verdadero.

conexión y un proceso con resultados.	
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Conseguir los resultados de un cliente sin conexión.	Se devuelve un valor nulo.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Conseguir los resultados de un cliente con conexión y un proceso sin resultados.	Se devuelve un valor nulo.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Conseguir los resultados de un cliente con conexión y un proceso con resultados.	Se devuelven los resultados más recientes del proceso.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Validar una dirección IP correcta.	Se devuelve un valor verdadero.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Validar una dirección IP incorrecta.	Se devuelve un valor falso.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Validar un puerto correcto.	Se devuelve un valor verdadero.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Validar un puerto incorrecto	Se devuelve un valor falso.
	<b>Resultado obtenido</b>
	Correcto

<b>Kaomi.Core</b>	
<b>Prueba</b>	<b>Resultado Esperado</b>
Descargar un archivo de una URI inexistente.	Se lanza una excepción.
	<b>Resultado obtenido</b>

	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Descargar un archivo de una URI existente.	El archivo se descarga correctamente.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Descargar un archivo sobrescribiendo el destino.	Se sobrescribe el archivo local con el archivo descargado.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Cargar un ensamblado inexistente.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Cargar un ensamblado incompatible.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Cargar un ensamblado al que le faltan referencias.	Se lanza una excepción.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Cargar un ensamblado correcto.	El ensamblado se carga en memoria.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Listar los ensamblados cargados.	Se devuelve una lista de ensamblados cargados.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Listar los ensamblados cuando no hay ninguno cargado.	Se devuelve una lista vacía.
	<b>Resultado obtenido</b>
	Correcto
<b>Prueba</b>	<b>Resultado Esperado</b>
Descargar un ensamblado que no está cargado.	El sistema no hace nada.
	<b>Resultado obtenido</b>

	Correcto
Prueba	Resultado Esperado
Descargar un ensamblado cargado.	El ensamblado se descarga de la memoria.
	Resultado obtenido
	Correcto
Prueba	Resultado Esperado
Instanciar un proceso de un ensamblado que no está cargado.	El sistema no hace nada.
	Resultado obtenido
	Correcto
Prueba	Resultado Esperado
Instanciar un proceso inexistente.	Se lanza una excepción.
	Resultado obtenido
	Correcto
Prueba	Resultado Esperado
Instanciar un proceso que ya está en ejecución.	Se lanza una excepción.
	Resultado obtenido
	Correcto
Prueba	Resultado Esperado
Instanciar un proceso que no está en ejecución.	El proceso se instancia correctamente.
	Resultado obtenido
	Correcto
Prueba	Resultado Esperado
Listar los procesos en ejecución.	Se devuelve una lista de procesos en ejecución.
	Resultado obtenido
	Correcto
Prueba	Resultado Esperado
Listar los procesos cuando ninguno está en ejecución.	Se devuelve una lista vacía.
	Resultado obtenido
	Correcto



## 7.2 Pruebas de Integración y del Sistema

Estos resultados corresponden a la ejecución final de las pruebas antes de la entrega del sistema.

Gestionar sensores: Ver distribución de sensores	
Prueba	Resultado Esperado
Abrir la aplicación sin tener ningún sensor <sup>1</sup> conectado.	El sistema muestra el mapa sin ningún icono.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Abrir la aplicación teniendo dos sensores en países diferentes.	El sistema muestra el mapa con dos iconos, uno en cada país.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Abrir la aplicación teniendo dos sensores en el mismo país.	El sistema muestra el mapa con un solo icono.
	<b>Resultado obtenido</b>
	Correcto

Gestionar sensores: Ver listado de sensores	
Prueba	Resultado Esperado
Ir al listado sin tener ningún sensor conectado.	El sistema muestra un listado vacío.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Ir al listado de todos los sensores.	El sistema muestra el listado con todos los sensores conectados.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Ir al listado desde uno de los iconos de mapa.	El sistema muestra el listado con los sensores de ese país.

<sup>1</sup> Durante el desarrollo de este proyecto, el cliente dispuso únicamente del sensor Gocator DLMI Technologies, por lo que es el único que se ha tenido en cuenta de forma explícita. No obstante, la arquitectura del sistema permite añadir los demás con modificaciones mínimas.

	<b>Resultado obtenido</b>
	Correcto

Gestionar sensores: Filtrar listado de sensores	
Prueba	Resultado Esperado
Filtrar un listado vacío.	El sistema sigue mostrando un listado vacío.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Aplicar un criterio cumplido por todos los sensores.	El sistema muestra el listado con todos los sensores.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Aplicar un criterio cumplido por alguno de los sensores.	El sistema muestra el listado con los sensores que cumplen el criterio.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Aplicar un criterio no cumplido por ningún sensor.	El sistema muestra un listado vacío.
	<b>Resultado obtenido</b>
	Correcto

Gestionar sensores: Modificar filtros de sensores	
Prueba	Resultado Esperado
Aplicar la modificación sin hacer ningún cambio.	El sistema muestra el mismo listado que antes.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Aplicar la modificación habiendo hecho cambios.	El sistema muestra un listado actualizado según los nuevos criterios.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Eliminar todos los filtros activos.	El sistema muestra un listado con todos los sensores conectados.
	<b>Resultado obtenido</b>
	Correcto

Gestionar sensores: Añadir un nuevo sensor	
Prueba	Resultado Esperado
Añadir un sensor de forma correcta.	El sistema tiene un sensor más que antes.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Introducir mal los valores de conexión.	El sistema muestra un error y no permite añadir el sensor.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Cancelar la operación antes de añadir un sensor.	El sistema no cambia el número de sensores conectados.
	<b>Resultado obtenido</b>
	Correcto

Gestionar sensores: Modificar un sensor existente	
Prueba	Resultado Esperado
Cambiar los valores de un sensor.	El sistema muestra los nuevos valores al ver los detalles de ese sensor.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Guardar los cambios manteniendo los valores anteriores.	El sistema muestra los valores originales del sensor.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Cancelar la modificación.	El sistema muestra los valores originales del sensor.
	<b>Resultado obtenido</b>
	Correcto

Gestionar sensores: Eliminar un sensor	
Prueba	Resultado Esperado
Aceptar la confirmación.	El sistema muestra un sensor menos que antes.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Denegar la confirmación.	El sistema muestra los mismos sensores que antes.
	<b>Resultado obtenido</b>

	Correcto
--	----------

Ver estado de un sensor	
Prueba	Resultado Esperado
Ver el estado de un sensor desconectado.	El sistema muestra los parámetros del sensor, pero no sus propiedades.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Ver el estado de un sensor conectado.	El sistema muestra el listado con todos los sensores conectados.
	<b>Resultado obtenido</b>
	Correcto

Modificar la configuración del sistema	
Prueba	Resultado Esperado
Aplicar la modificación sin hacer ningún cambio.	El sistema se reinicia y muestra los mismos valores de configuración que antes.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Aplicar la modificación habiendo hecho cambios.	El sistema se reinicia y muestra los nuevos valores de configuración.
	<b>Resultado obtenido</b>
	Correcto
Prueba	Resultado Esperado
Cancelar la modificación.	El sistema no se reinicia y mantiene los mismos valores de configuración.
	<b>Resultado obtenido</b>
	Correcto

## 7.3 Pruebas de Usabilidad

Aquí irá el resultado de las pruebas de usabilidad, incluyendo copias rellenas del cuestionario que aparece en el apartado de diseño de las pruebas.

### 7.3.1 Usuario de pruebas 1

¿Usa un ordenador frecuentemente?
Todos los días
¿Qué tipo de actividades realiza con el ordenador?
Es parte de mi trabajo o profesión
¿Ha usado alguna vez software como el de esta prueba?
No, aunque si empleo otros programas que me ayudan a realizar tareas similares
¿Qué busca principalmente en un programa?
Que tenga todas las funciones necesarias

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Comprende la estructura de navegación de la aplicación?	X			
¿En algún momento siente que debería recibir ayuda?				X
¿Le resulta sencillo el uso de la aplicación?		X		
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿En algún momento el sistema actúa de forma inesperada?				X
¿El tiempo de respuesta de la aplicación es muy grande?				X
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
Los textos de la interfaz son...		X		
Los iconos e imágenes son...	X			
Los colores empleados son...		X		
Diseño de la Interfaz	Si		No	A veces
¿Le resulta fácil de usar?	X			
¿El diseño de las pantallas es claro y atractivo?	X			
¿Cree que el programa está bien estructurado?	X			

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	Correcto
<i>El usuario no necesita demasiado tiempo para cada tarea</i>	Correcto
<i>El usuario comete errores leves</i>	No
<i>El usuario comete errores graves</i>	No
<i>El usuario navega de forma innecesaria</i>	El usuario prefiere no utilizar el acceso rápido, y navegar mediante los controles normales.
<i>El usuario parece desorientado en cuanto a la estructura de la aplicación</i>	No

## 7.3.2 Usuario de pruebas 2

¿Usa un ordenador frecuentemente?
Todos los días
¿Qué tipo de actividades realiza con el ordenador?
Es parte de mi trabajo o profesión
¿Ha usado alguna vez software como el de esta prueba?
Sí, he empleado software similar
¿Qué busca principalmente en un programa?
Que sea intuitivo

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Comprende la estructura de navegación de la aplicación?</i>	X			
<i>¿En algún momento siente que debería recibir ayuda?</i>				X
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿En algún momento el sistema actúa de forma inesperada?</i>				X
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>Los textos de la interfaz son...</i>	X			

Los iconos e imágenes son...	X			
Los colores empleados son...		X		
Diseño de la Interfaz		Si	No	A veces
¿Le resulta fácil de usar?		X		
¿El diseño de las pantallas es claro y atractivo?				X
¿Cree que el programa está bien estructurado?		X		
Observaciones				
El usuario indica que la página de detalles de un sensor tiene una estructura confusa y presenta la información de forma poco organizada.				

Aspecto Observado	Notas
El usuario comienza a trabajar de forma rápida por las tareas	Correcto
El usuario no necesita demasiado tiempo para cada tarea	Correcto
El usuario comete errores leves	No
El usuario comete errores graves	No
El usuario navega de forma innecesaria	No
El usuario parece desorientado en cuanto a la estructura de la aplicación	No

### 7.3.3 Usuario de pruebas 3

¿Usa un ordenador frecuentemente?
Todos los días
¿Qué tipo de actividades realiza con el ordenador?
Es parte de mi trabajo o profesión
¿Ha usado alguna vez software como el de esta prueba?
No, nunca
¿Qué busca principalmente en un programa?
Que tenga todas las funciones necesarias

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Comprende la estructura de navegación de la aplicación?</i>		X		
<i>¿En algún momento siente que debería recibir ayuda?</i>				X
<i>¿Le resulta sencillo el uso de la aplicación?</i>	X			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿En algún momento el sistema actúa de forma inesperada?</i>				X
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				X
Calidad de la Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>Los textos de la interfaz son...</i>	X			
<i>Los iconos e imágenes son...</i>	X			
<i>Los colores empleados son...</i>	X			
Diseño de la Interfaz		Si	No	A veces
<i>¿Le resulta fácil de usar?</i>		X		
<i>¿El diseño de las pantallas es claro y atractivo?</i>		X		
<i>¿Cree que el programa está bien estructurado?</i>		X		
Observaciones				
El usuario muestra sorpresa al descubrir el acceso rápido, pero está de acuerdo en su utilidad.				
Aspecto Observado	Notas			
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	Correcto			
<i>El usuario no necesita demasiado tiempo para cada tarea</i>	Correcto			
<i>El usuario comete errores leves</i>	No			
<i>El usuario comete errores graves</i>	No			
<i>El usuario navega de forma innecesaria</i>	No			
<i>El usuario parece desorientado en cuanto a la estructura de la aplicación</i>	No			



## 7.3.4 Observaciones

Tras las pruebas de usabilidad, se obtienen comentarios indicando que la página de detalles de un sensor tiene una distribución poco óptima. Esto se traduce en una modificación de dicha página, que queda con un aspecto como este:



*Figura 7.1: Pantalla de detalles de un sensor.*

## 7.4 Pruebas de Rendimiento

Tras ejecutar las pruebas de rendimiento, se obtienen los siguientes resultados:

- A pesar de estar ejecutándose sobre el mismo ordenador, los 30 servidores web se inician y lanzan su proceso correspondiente sin diferencias apreciables de tiempo.
- Además, el rendimiento del resto del sistema no se ve afectado por la presencia de los servidores.
- La aplicación es capaz de conectar de forma secuencial todos los sensores sin diferencias de rendimiento.
- Las pantallas que deben mostrar datos de varios servidores, como la vista de listado, funcionan de forma ágil, incluyendo operaciones como los filtrados, que requieren trabajar sobre la lista completa de sensores.
- El hecho de que la aplicación esté haciendo peticiones periódicas a 30 servidores no provoca diferencias de rendimiento en el resto del sistema.

Cabe destacar que las condiciones de las pruebas de rendimiento son más duras de lo que se espera durante la ejecución real del sistema:

- En circunstancias normales, solamente habrá un servidor web en cada ordenador remoto, en vez de 30.
- De forma similar, es probable que el ordenador que ejecute la aplicación no tenga que llevar a cabo ninguna otra tarea de forma simultánea.

### 7.4.1 Conclusiones

Teniendo en cuenta las condiciones y resultados de las pruebas de rendimiento, es razonable concluir que tanto la aplicación local como los servidores remotos son capaces de ejecutarse con un rendimiento óptimo en la práctica totalidad de circunstancias de funcionamiento.

## Capítulo 8. Manuales del Sistema

### 8.1 Manual de Instalación

La instalación de este sistema comprende dos pasos principales: configurar los servidores remotos e instalar la aplicación local. Cada uno de estos pasos puede completarse por separado, ya que no son dependientes entre sí.

#### 8.1.1 Instalación del servidor remoto

Este paso debe repetirse para cada uno de los ordenadores en los que se encuentre un sensor remoto.

Antes de empezar, se debe comprobar que el ordenador tenga instalada la plataforma **.NET Framework**, con una versión mínima **4.6.1**.

Además, el tipo de sensor que se desea conectar debe tener un módulo adaptador desarrollado específicamente para él. Cabe destacar que los módulos adaptadores no son normalmente compatibles con otros tipos de sensor.

Una vez comprobados estos requisitos, se puede proceder con la instalación.

##### 8.1.1.1 Preparar e iniciar el servidor

El primer paso es copiar los archivos correspondientes tanto al servidor como al módulo adaptador a una carpeta del ordenador donde no se requieran permisos de escritura. Una buena ubicación es la carpeta de documentos, por ejemplo.

La distribución incluye todas las dependencias necesarias para ejecutar módulos adaptadores compatibles con las plataformas de destino **net461**, **netstandard2.0** y **netcoreapp2.2**.

A partir de este momento, el servidor puede iniciarse ejecutando el archivo **Kaomi.WebAPI.exe**.

Hay dos posibilidades para poner en ejecución el proceso de sensor. Ambas dan el mismo resultado, por lo que se debe elegir una de ellas.

### 8.1.1.2 Opción A: Instanciar el proceso manualmente

Esta opción debe llevarse a cabo con el servidor en ejecución, y deberá repetirse cada vez que el servidor se detenga.

Para ello, basta con abrir un navegador y ejecutar las siguientes URI:

- `http://localhost:5000/Kaomi/Load?path=SensorAssembly.dll`, donde **5000** es el puerto de ejecución (indicado en la consola del servidor) y **SensorAssembly.dll** es el ensamblado que contiene el proceso de sensor.
- `http://localhost:5000/Kaomi/InstanceProcess?id=SensorAssembly&type=SensorProcess`, donde **SensorAssembly** es la ID del ensamblado (que se muestra como resultado de la operación anterior) y **SensorProcess** es el nombre de la clase que implementa el proceso de sensor.

En este punto el proceso de sensor debería estar ejecutándose, y la instalación del servidor está completa.

### 8.1.1.3 Opción B: Establecer proceso de inicio automático

Esta opción debe llevarse a cabo con el servidor detenido, y una vez completada debería funcionar, aunque el servidor se detenga y se inicie de nuevo.

Para ello, se debe abrir el archivo **Kaomi.xml**, y modificar su contenido para que quede de la siguiente manera:

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration Format="A">
  <Startup>
    <Setting Key="SensorAssembly">SensorProcess</Setting>
  </Startup>
</Configuration>
```

**SensorAssembly** corresponde al nombre del ensamblado que contiene el proceso de sensor, y **SensorProcess** a la clase específica que implementa dicho proceso.

Si se inicia el servidor a partir de ahora, debería instanciar automáticamente el proceso de sensor, lo que concluye esta instalación.

## 8.1.2 Instalación de la aplicación local

Antes de instalar la aplicación local, es necesario asegurarse de que el ordenador está ejecutando **Windows 10** con una versión igual o posterior a la **1803**.

Además, la carga de aplicaciones desde orígenes diferentes a la tienda debe estar activada. Para ello se pueden seguir los siguientes pasos:

- Abrir la [Configuración](#).
- Entrar en la sección [Actualización y Seguridad](#).
- En la barra lateral, ir al apartado [Para programadores](#).
- El valor del ajuste [Usar las opciones de programador](#) debe establecerse como [Modo de Programador](#), aceptando todas las confirmaciones que necesite el sistema.

Una vez hecho esto, basta con acceder a la carpeta que contiene los archivos de instalación de la aplicación local y ejecutar el archivo [Add-AppDevPackage.ps1](#) con derechos de administrador.

Este script se encargará de instalar la aplicación y todas las dependencias necesarias.

## 8.2 Manual de Usuario

Una vez que todos los componentes necesarios están instalados, se puede empezar a usar la aplicación.

### 8.2.1 Primer uso

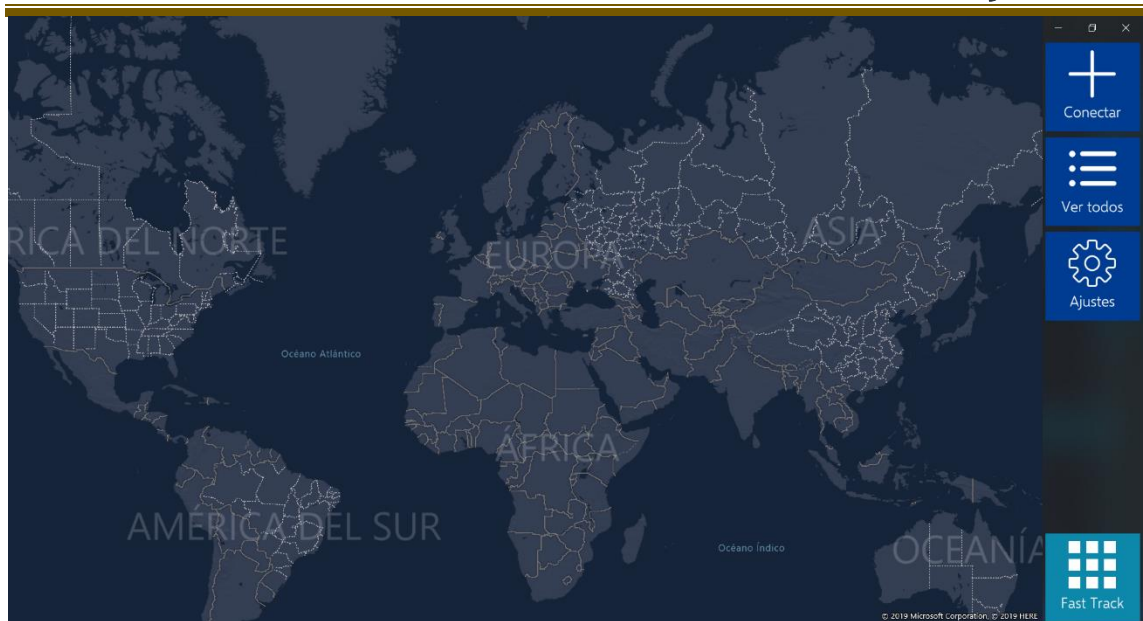
La primera vez que se ejecute la aplicación, se mostrará la Experiencia de Nuevo Usuario:



*Figura 8.1. Experiencia de Nuevo Usuario*

Esta pantalla solo se mostrará en esta ocasión, y permite introducir a nuevos usuarios a la aplicación, además de permitir importar una configuración anterior en caso de haber restaurado los ajustes.

En este caso, vamos a simular un uso desde cero, por lo que pulsando el botón Saltar se llegará a la pantalla principal.



*Figura 8.2. Vista inicial de la página principal*

Por ahora no se muestra ningún icono en el mapa, ya que no hay ningún sensor conectado. Para conectar un sensor, se puede pulsar sobre el botón Conectar, que llevará a la pantalla de conexión de nuevos sensores:



*Figura 8.3. Paso 1 de la conexión de nuevos sensores*

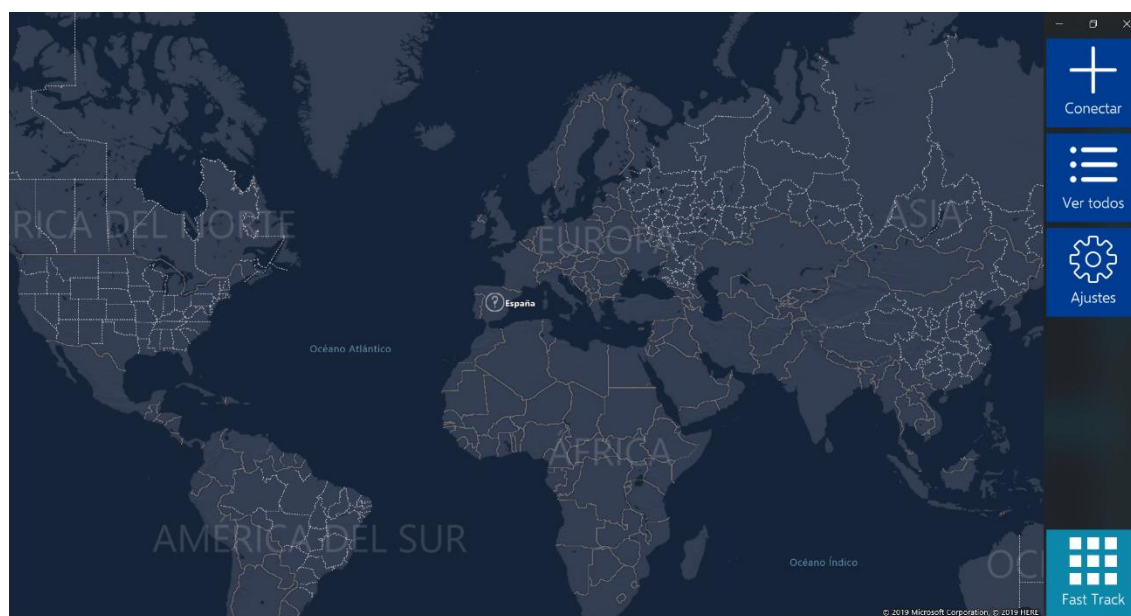
Esta pantalla muestra una secuencia de pasos necesarios para conectar sensores al sistema. Tras introducir los datos necesarios en el primer paso, el sistema mostrará el segundo paso:

The screenshot shows a web application window titled 'Algo más de información'. At the top, there are three tabs: 'Conexión', 'Identificación', and 'Datos adicionales', with 'Identificación' being the active tab. Below the tabs, a message states: '¡Estupendo! El servidor remoto parece estar funcionando correctamente.' followed by 'Ahora, necesitamos algunos datos más para saber dónde colocar el nuevo sensor. La mayoría de estos datos pueden ser modificados más adelante.' On the right side, there are two buttons: 'Cancelar' (with a close icon) and 'Continuar' (with a right arrow icon). The main form area contains several input fields: 'Nombre del sensor' (text input with placeholder 'Nuevo sensor sin nombre'), 'País de origen' (dropdown menu with 'Indefinido'), 'Ciudad' (text input), 'Lugar de instalación' (dropdown menu with 'Indefinido'), 'Operaciones' (dropdown menu with 'Indefinido'), and 'Comentarios' (text area). In the bottom right corner, there is a 'Fast Track' button with a grid icon.

*Figura 8.4. Paso 2 de la conexión de nuevos sensores*

En algunos casos, podría ser necesario introducir datos adicionales. No obstante, los tipos de sensor considerados en la versión actual no lo necesitan, por lo que se pueden conectar directamente.

Una vez terminado este proceso, el sistema mostrará de nuevo la pantalla principal:



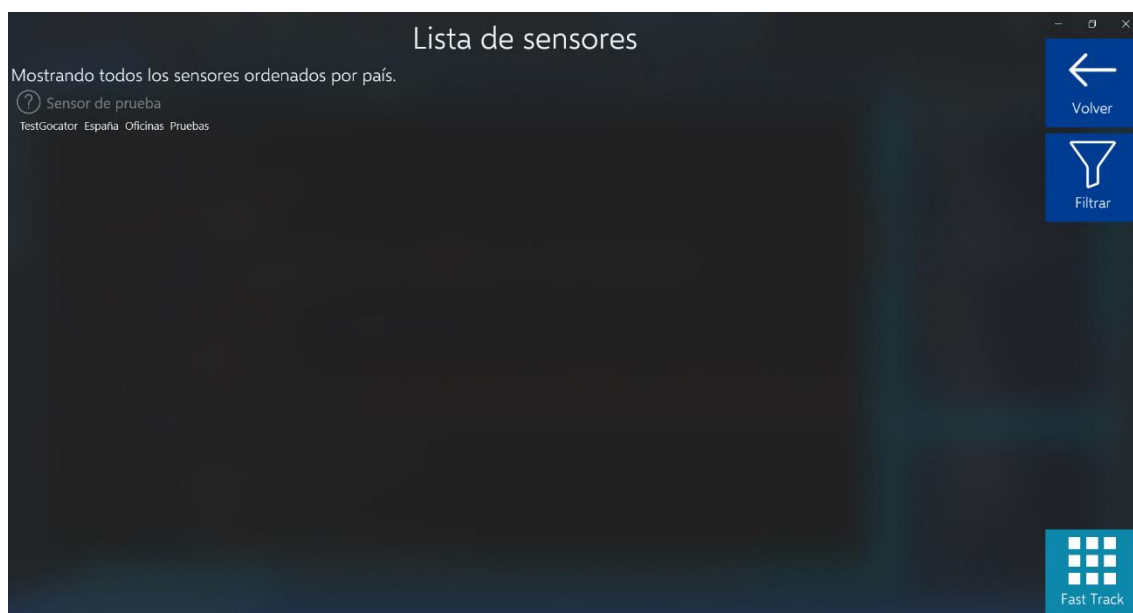
*Figura 8.5. Página principal con un sensor conectado*

Ahora se puede ver un icono sobre el mapa, correspondiente al sensor que acaba de ser añadido.



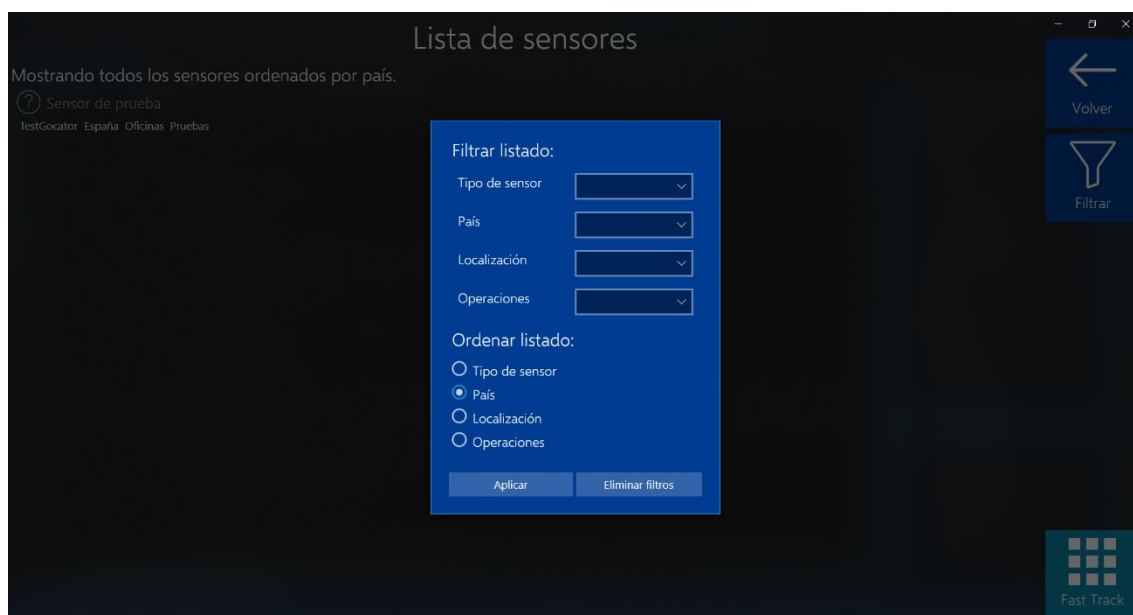
## 8.2.2 Funciones principales

Para ver el listado de sensores, se puede pulsar sobre el botón Ver todos, o bien sobre uno de los iconos. En el primer caso, se mostrarán todos los sensores conectados al sistema. En el segundo, solo aquellos pertenecientes al país sobre el que se ha pulsado:



*Figura 8.6. Listado inicial de sensores*

Este listado puede modificarse en cualquier momento pulsando sobre el botón Filtrar, que abrirá el diálogo de filtros:



*Figura 8.7. Diálogo de filtros y ordenación*

Pulsando sobre cualquiera de los sensores que aparecen en el listado, el sistema mostrará sus detalles. Si el sensor está desconectado, la pantalla tendrá este aspecto:



*Figura 8.8. Detalles de un sensor desconectado*

En otro caso, se mostrarán detalles adicionales:



*Figura 8.9. Detalles de un sensor conectado*

Desde esta pantalla es posible modificar los datos del sensor, pulsando sobre el botón Modificar:

**Modificar sensor**

Nombre:  Nombre que se mostrará asociado al sensor.

Dirección IP:  Punto de conexión al servidor remoto donde está alojado el sensor.

Puerto:  Punto de conexión al servidor remoto donde está alojado el sensor.

Tipo de sensor:  Indica de qué tipo es el sensor (y qué proceso modular debe usarse).

País:  Usado con fines de filtrado y clasificación en la interfaz.

Localización:  Usado con fines de filtrado y clasificación en la interfaz.

Ciudad:  Usado únicamente para facilitar la identificación.

Operaciones:  Usado con fines de filtrado y clasificación en la interfaz.

Comentarios:  Texto arbitrario para añadir cualquier otro detalle útil.

Cancelar

Guardar

Fast Track

Figura 8.10. Modificación de los datos de un sensor

## 8.2.3 Acceso Rápido

Desde cualquier pantalla, es posible entrar en el sistema de Acceso Rápido pulsando el botón correspondiente en la esquina inferior derecha de la pantalla:

**Lista de sensores**

Mostrando todos los sensores ordenados por país.

TestGocator España Oficinas Pruebas

**Acceso Rápido**

Este es el sistema de Acceso Rápido. A través de la cuadrícula podrás llegar a todas las opciones de la aplicación.

Si en algún momento te pierdes, solo tienes que hacer clic derecho sobre el botón central para cerrar el panel.

Recuerda que también puedes usar el teclado numérico para navegar.

Sensores

Ajustes

Listados

Volver

Filtrar

Fast Track

Figura 8.11. Panel de Acceso Rápido

Esta interfaz permite navegar rápidamente por la aplicación siguiendo las instrucciones que aparecen en el lado derecho.

Además, es posible utilizar atajos de teclado para moverse por el sistema:

- F1 o 0 del teclado numérico: Abrir el panel de Acceso Rápido
- 0 del teclado numérico: Cerrar el panel de Acceso Rápido (si ya está abierto)
- 7, 8, 9, 4, 6, 1, 2, y 3 del teclado numérico: Pulsar la opción correspondiente de la cuadrícula de botones.
- 5 del teclado numérico: Pulsar la opción central de la cuadrícula de botones.

## 8.3 Manual del Programador

A continuación se indican los pasos necesarios para adaptar la aplicación a un nuevo tipo de sensor, como podría ser alguno de los tipos estudiados en los aspectos teóricos que finalmente no fueron implementados.

Conceptualmente se debe tener en cuenta que existen dos tipos de propiedades que se pueden mostrar en la interfaz de usuario: textuales y numéricas.

Las propiedades textuales muestran un nombre y un valor, y son útiles para indicar estados, modos de funcionamiento o versiones de firmware.

Las propiedades numéricas muestran una etiqueta y valores numéricos para mínimo, máximo y valor actual, y son útiles para indicar frecuencias, temperaturas u otros valores cuantificables.

### 8.3.1 Crear un proceso modular nuevo

Los procesos modulares se crean en forma de clase C#. Para ello, el primer paso es crear un proyecto nuevo, o abrir uno ya existente.

Normalmente los procesos suelen implementarse como bibliotecas de clases, aunque puede usarse una aplicación de consola para ayudar durante la depuración. Si el proyecto tiene un punto de entrada estándar (como un método Main en proyectos de console) es importante darse cuenta de que ese punto de entrada solo sirve para hacer pruebas durante el desarrollo, y no será llamado por el servidor una vez que el proceso se ponga en ejecución.

En cualquier caso, el proyecto debe incluir una dependencia para Kaomi.Core.

A continuación, se debe crear una clase nueva que extienda de la clase abstracta `Kaomi.Core.Model.KaomiProcess`.

Tras implementar los métodos abstractos requeridos, se debería obtener una clase con un aspecto similar al siguiente:

```
using System;
using Kaomi.Core.Model;

namespace CustomAssembly
{
    public class CustomProcess: KaomiProcess
    {
        public override void OnInitialize()
        {

        }

        public override void OnIteration()
        {

        }

        public override void OnUserMessage(string message)
        {

        }

        public override void OnFinalize()
        {

        }
    }
}
```

Cada uno de los métodos tiene un propósito diferenciado:

- **OnInitialize:** Se ejecuta una vez cuando el proceso es cargado en memoria. En este método se pueden incluir tareas como leer configuración, abrir la conexión al sensor o preparar algún otro atributo necesario. A efectos prácticos sustituye a un constructor tradicional.
- **OnIteration:** Se ejecuta una vez por cada iteración, y debería usarse para conseguir un nuevo conjunto de resultados desde el sensor. Es importante destacar que este método no debería contener bucles de lectura, ya que el servidor se encargará de llamarlo cada vez que sea necesario (es decir, la ejecución debe terminar de forma más o menos ágil).
- **OnUserMessage:** Si este proceso admite mensajes del usuario, la respuesta a estos mensajes debe incluirse aquí. En otro caso, la implementación debería dejarse vacía. La comprobación de mensajes de usuario se hace justo después de ejecutar cada iteración, y antes de empezar el tiempo de espera.
- **OnFinalize:** Se ejecuta una vez cuando el proceso va a descargarse de memoria. En este punto es conveniente liberar recursos como archivos o conexiones de red. Cabe destacar que una vez que se alcanza este método, un proceso no puede cancelar su propia eliminación.

Una implementación de referencia podría ser la siguiente:

```
using System;
using Kaomi.Core.Model;
using ThirdParty.Sensor.SDK;

namespace CustomAssembly
{
    public class CustomProcess: KaomiProcess
    {
        private ThirdPartySensor _sensor;

        public override void OnInitialize()
        {
            _sensor = new ThirdPartyTemperatureSensor("127.0.0.1:4242", model:3);
            base.IterationDelay = TimeSpan.FromMinutes(1);
        }

        public override void OnIteration()
        {
            ThirdPartySensorResults results = _sensor.GetResults();
            var stringResults = $"[Version={results.Version}&Calibrated={results.Calibrated}] " +
                                $"| Temperature=0={results.Temperature}=50] ";

            base.NotifyResults(stringResults);
        }

        public override void OnUserMessage(string message)
        {
            if (message.ToUpperInvariant().Equals("EXIT"))
                base.RequestFinalization = true;
        }

        public override void OnFinalize()
        {
            _sensor.Dispose();
        }
    }
}
```

Cabe destacar el formato necesario para que los resultados sean comprensibles desde la interfaz:

**[Propiedades textuales|Propiedades numéricas]**

Donde cada propiedad textual está separada de las demás por un &, y tiene el siguiente formato:

**Nombre=Valor**

Y donde cada propiedad numérica está separada de las demás por un &, y tiene el siguiente formato:

**Nombre=Mínimo=Valor actual=Máximo**

Otros detalles son las llamadas a propiedades de la clase KaomiProcess:

- **base.IterationDelay:** Obtiene o establece un objeto TimeSpan que indica el tiempo de espera entre iteraciones. Nota: Si el tiempo de espera es muy pequeño o grande, el servidor podría ignorarlo y establecer un valor razonable. Se recomiendan valores entre diez segundos y diez minutos.
- **base.NotifyResults:** Envía un resultado al servidor para que lo ponga a disposición de la interfaz de usuario. La cola de resultados no es infinita, sino que tiene un tamaño máximo. Si este tamaño se alcanza, el servidor empezará a eliminar los resultados más antiguos.
- **base.RequestFinalization:** Si esta propiedad se establece a true, el proceso no ejecutará ninguna iteración adicional, y el servidor lo descargará lo antes posible.



## Capítulo 9. Conclusiones y otras Ampliaciones

### 9.1 Conclusiones

Llegados a este punto, se ha desarrollado un sistema que cumple los objetivos planteados al inicio. En ese aspecto, por lo tanto, el proyecto puede considerarse exitoso.

Nota: El resto de la conclusión que aparece en la versión entregable del documento no tiene utilidad práctica, y solo está para llenar hueco.

### 9.2 Ampliaciones

El sistema desarrollado podría ser ampliado de estas formas:

#### 9.2.1 Mejorar las capacidades del mapa

Una posible mejora sería ampliar las capacidades del mapa que se muestra en la página principal. Por ejemplo:

- Zoom dinámico, de forma que inicialmente se muestre un icono por país, pero si el usuario amplía lo suficiente un país, se pasa a mostrar iconos individuales de los sensores de ese país según su ciudad.
- Permitir filtrar los iconos que se muestran en el mapa según criterios significativos.

#### 9.2.2 Registrar notificaciones

El sistema podría ampliarse de forma que las notificaciones emitidas cuando un sensor cambia de estado queden registradas en algún lugar, como bases de datos, un fichero de registro o algún otro tipo de interfaz de almacenamiento.

#### 9.2.3 Internacionalizar la aplicación

Podría ser interesante traducir las cadenas de texto para que la aplicación sea utilizable por personas que no entiendan el español.



# Capítulo 10. Referencias Bibliográficas

## 10.1 Referencias en Internet

Páginas Web consultadas para cualquier aspecto relacionado con el desarrollo del sistema o su documentación.

**Paul Johnson [Microsoft]:** ¿Qué es una aplicación para la Plataforma Universal de Windows (UWP)?

<https://docs.microsoft.com/es-es/windows/uwp/get-started/universal-application-platform-guide>. 2019.

**Bill Wagner [Microsoft]:** Creación de un cliente REST con .NET Core.

<https://docs.microsoft.com/es-es/dotnet/csharp/tutorials/console-webapiclient>. 2019

**Serena Zheng [Microsoft]:** Send a local toast notification.

<https://docs.microsoft.com/en-us/windows/uwp/design/shell/tiles-and-notifications/send-local-toast> 2019

**Microsoft Design Team:** Fluent Design System.

<https://www.microsoft.com/design/fluent>. 2019

**Carlos Lubián Fernández:** Guía de Estilo Flushing.

<http://flushing.carlubian.es>