

**Sprint final**  
**Framework para facilitar la**  
**reutilización en la construcción**  
**de sistemas de medición**



**Presentado a:**

Julio A. Hurtado

**Sprint 1**

**Presentado por:**

Juan Sebastián Osorio AguileraJuan

Carlos Neuta MontenegroDiego

Andrés Vidal López

Andrea Cuatindioy Ortiz

Manuel López Valencia

**Facultad de Ingeniería Electrónica y Telecomunicaciones (FIET)**

**Programa de Ingeniería de sistemas**

**Popayán, agosto del 2022**

**Resumen:** Los sistemas de medición son una clase de sistemas utilizados para medir los valores relevantes de un proceso o un producto. El sistema de medición se usa para el control de calidad en productos producidos que luego, se pueden usar para separar los productos inaceptables o para categorizar los productos en función de su calidad. En algunos sistemas, los resultados de las mediciones se almacenan, en el caso de que, en el futuro, surja la necesidad de requerir esta información, p. ej. si los clientes se quejan sobre los productos que pasaron la prueba del sistema de medición.

Aunque un sistema de medición contiene cantidades considerables de software, una parte sustancial de estos sistemas es Hardware, ya que está conectado al mundo real a través de una serie de sensores (p.e. un sensor de movimiento) y actuadores (p.e. un motor). Los sensores proporcionan información sobre el mundo real a través de los impulsos leídos. Con respecto a los actuadores, a menudo solo necesita pasarles un valor establecido expresado en conceptos de dominio de aplicación como por ejemplo velocidad angular o fuerza.

Un Sistema de Medición, sin embargo, es más que sensores y actuadores. Comienza un ciclo de medición típico con un disparador que indica que un producto, o elemento de medición, está ingresando al sistema.

## Representación de la arquitectura Vista de Requerimientos

### Minimum Viable Product (MVP)

#### Sprint II

#### Caso de uso A01

Caso de Uso	Ingreso de los elementos	Identificador: A01
Actores	Tipos de usuario: Sistema	
Tipo	<i>Primario</i>	
Referencias		
Precondición	El sistema debe tener acceso a la base de datos. El usuario administrativo debe estar ingresado en el sistema y tener un perfil administrativo.	
Postcondición	El sistema deberá presentar una pantalla de confirmación por cada registro realizado. Toda la información correspondiente al registro se almacenará.	
Descripción	Al realizar el ingreso de los elementos el sistema va a solicitar cierta información, una vez se digitan los datos correspondientes se corrobora si este elemento ya se encuentra registrado y en caso de ser así envía un mensaje reportando esta inconsistencia, de lo contrario procede a realizar el respectivo registro.	
Comentarios	En la plataforma existirán campos que serán obligatorios al momento de ingresar los datos. Los datos solicitados pueden variar dependiendo el tipo de elemento.	

#### Curso Normal

Nro.	Ejecutor	Paso o Actividad	
1	Sistema	El <b>sistema</b> debe seleccionar la opción para realizar el registro	
		1.1	Usuarios: Sistema

## Cursos Alternos

Nro.	Descripción de acciones alternas	
1	El sistema muestra en pantalla mensajes de error asociados a diferentes errores cometidos por el sistema cuando ingrese los elementos, como los siguientes:	
	1.1	Campos vacíos.
	1.2	Números negativos.
	1.3	Caracteres no alfanuméricos.

## Caso de uso A02

Caso de Uso	Recolección de datos por parte de los sensores	<b>Identificador:</b> A02
Actores	Tipos de usuario: Sistema, Sensores	
Tipo	<i>Primario</i>	
Referencias	A01	
Precondición	Al momento de recolectar los datos el sistema debe verificar que se encuentra ingresado el elemento. El sistema debe tener acceso a los datos.	
Postcondición	El sistema deberá presentar una pantalla de confirmación por cada registro realizado y mostrar que los datos fueron recolectados de manera exitosa. Toda la información correspondiente al registro se almacenará.	
Descripción	Al realizar la recolección de los datos el sistema mide las diversas variables relevantes del elemento de medición, por ejemplo; el peso, el tamaño, la forma, etc, Una vez recolectados los datos, el sistema debe mandar un mensaje de éxito.	
Comentarios	La información puede variar según el tipo de elemento	

## Curso Normal

Nro.	Ejecutor	Paso o Actividad	
1	Sistema	El <b>sistema</b> debe seleccionar la opción para realizar el registro	
		1.1	Usuarios: Sistema
		1.2	Sensor

## Cursos Alternos

Nro.	Descripción de acciones alternas	
1	El sistema muestra en pantalla mensajes de error asociados a diferentes errores cometidos por el usuario cuando digita los datos, como los siguientes:	
	1.1	Campos vacíos.
	1.2	Elemento incorrecto.

## Caso de uso A03

Caso de Uso	Categorizar los elementos de medición	<b>Identificador:</b> A03
Actores	Tipos de usuario: Sistema, Elementos de medición	
Tipo	<i>Primario</i>	
Referencias	A02	
Precondición	Al momento de recolectar los elementos de medición el sistema debe transformar los datos para realizar la comparación con los valores ideales o umbrales para poder tomar decisiones. El sistema debe tener acceso a los datos.	
Postcondición	El sistema deberá mostrar la clasificación de los elementos según los valores obtenidos	
Descripción	Al realizar la categorización los elementos se clasifican de acuerdo a su información.	
Comentarios	La información puede variar según el tipo de elemento.	

## Curso Normal

Nro.	Ejecutor	Paso o Actividad	
1	Sistema	El <b>sistema</b> compara los valores obtenidos por medio de los sensores con los valores:	
		1.1	Valor ideal
		1.2	Valor umbral
2	Elementos de medición	El elemento de medición se categoriza a partir de los resultados obtenidos en las comparaciones	

## Cursos Alternos

Nro.	Descripción de acciones alternas	
1	El sistema muestra en pantalla mensajes de error asociados a diferentes errores cometidos por el usuario cuando digita los datos, como los siguientes:	
	1.1	Campos vacíos.
2	El sistema debe comprobar que los valores sean correctos o que coincidan con los elementos físicos para evitar errores en la clasificación de los elementos.	

## Caso de uso A04

Caso de Uso	Acción de acuerdo a su categoría	<b>Identificador:</b> A04
Actores	Tipos de usuario: Sistema, Elementos de medición, Actuador	
Tipo	<i>Primario</i>	
Referencias	A02, A03	
Precondición	Al momento de tomar acción sobre los elementos según su categoría el sistema debe poder rechazar o imprimir la clasificación en el artículo para que posteriormente pueda ser identificado automáticamente. El sistema debe tener acceso a los datos.	
Postcondición	El sistema deberá mostrar la clasificación de los elementos según los valores obtenidos.	
Descripción	Al realizar la acción, los elementos categorizados serán verificados de acuerdo a la información y se comparan los valores para ver si se aceptan o se rechazan.	
Comentarios	No aplica.	

## Curso Normal

Nro.	Ejecutor	Paso o Actividad	
1	Sistema	El <b>sistema</b> compara los valores obtenidos por medio de los sensores con los valores:	
		1.1	Valor ideal
		1.2	Valor umbral

2	Elementos de medición	El elemento de medición se categoriza a partir de los resultados obtenidos en las comparaciones
3	Actuador	El actuador toma acción sobre los elementos ingresados y decide si los acepta o los rechaza

### **Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>	
1	El sistema muestra en pantalla mensajes de error asociados a diferentes errores cometidos por el usuario cuando digita los datos, como los siguientes:	
	1.1	Campos vacíos.
2	El sistema debe comprobar que los valores sean correctos o que coincidan con los elementos físicos para evitar errores en la clasificación de los elementos.	
3	El sistema debe poder ingresar o rechazar los elementos medidos	

## Escenarios de calidad (atributos de calidad)

Cod. Escenario: EAC01	
<b>Descripción</b>	
El sistema debe estar dispuesto a posibles cambios, así como también en agregaciones a su funcionalidad de forma que no afecte el código general.	
Interesado: Cliente	Atributo: Modificabilidad
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	La necesidad de un sistema flexible que pueda responder correctamente a las necesidades que puedan surgir en respuesta a nuevas problemáticas.
<b>Estímulo</b>	Diseño de arquitectura y modelo de software.
<b>Entorno</b>	En desarrollo
<b>Artefacto</b>	Proyecto
<b>Respuesta</b>	Un diseño robusto, coherente y flexible para el proyecto
<b>Medida de respuesta</b>	N/A

Cod. Escenario: EAC02	
<b>Descripción</b>	
El sistema debe soportar el tráfico masivo junto a las peticiones de manera correcta y usando adecuadamente los recursos para un correcto funcionamiento.	
Interesado: Usuario	Atributo: Desempeño de Visualización
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	La correcta visualización y uso de la aplicación para todos los usuarios interesados.
<b>Estímulo</b>	Visualización, consulta
<b>Entorno</b>	En ejecución
<b>Artefacto</b>	Aplicación



<b>Respuesta</b>	Datos ordenados y legibles sobre las condiciones especificadas
<b>Medida de respuesta</b>	< 5s

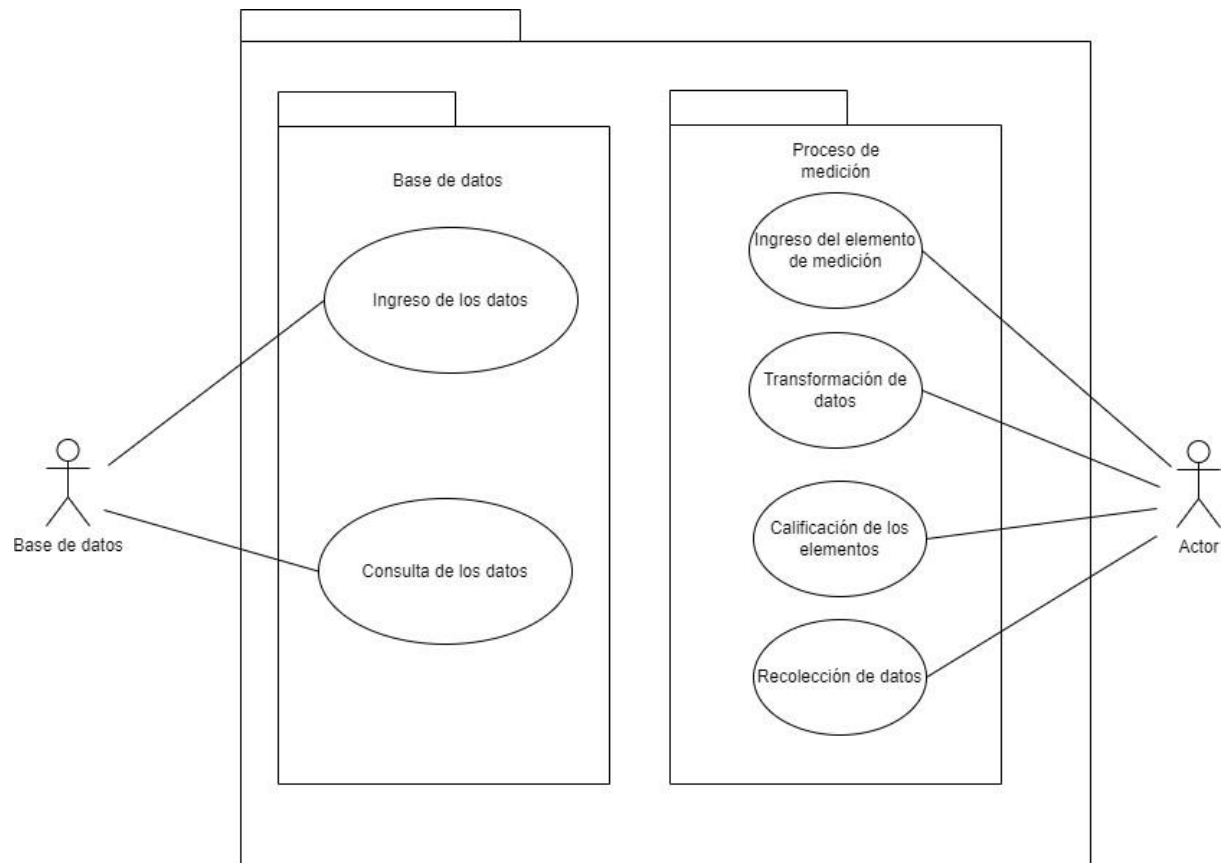
Cod. Escenario: EAC03	
<b>Descripción</b>	
<b>Las peticiones a la base de datos deben realizarse de manera eficaz sin entorpecer los tiempos estimados de respuesta.</b>	
Interesado: Usuario	Atributo: Desempeño de registro
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	Una capacidad de respuesta activa ante peticiones realizadas por el usuario.
<b>Estímulo</b>	Peticiones
<b>Entorno</b>	En ejecución
<b>Artefacto</b>	Aplicación, Base de datos
<b>Respuesta</b>	Una emisión rápida que permita una actualización eficiente de la información suministrada.
<b>Medida de respuesta</b>	< 2s

Cod. Escenario: EAC04	
<b>Descripción</b>	
<b>El sistema debe contar con un refresco oportuno de nuevas actualizaciones en el registro de problemas para todos los usuarios.</b>	
Interesado: Usuario	Atributo: Desempeño de refresco
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	El sistema debe ser concurrente para los usuarios.
<b>Estímulo</b>	Registros
<b>Entorno</b>	En ejecución
<b>Artefacto</b>	Aplicación, Base de datos

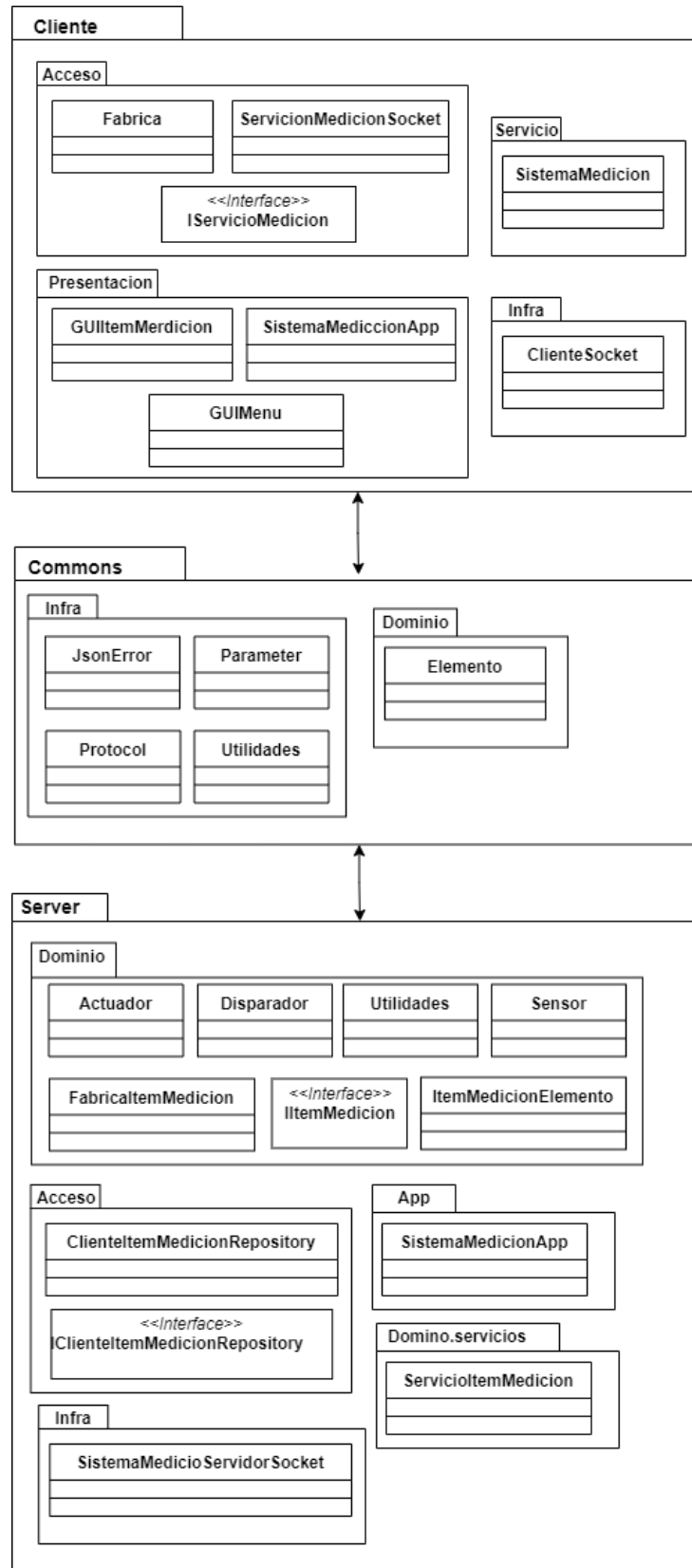
<b>Respuesta</b>	Una actualización inmediata de los cambios realizados en el sistema.
<b>Medida de respuesta</b>	< 1s

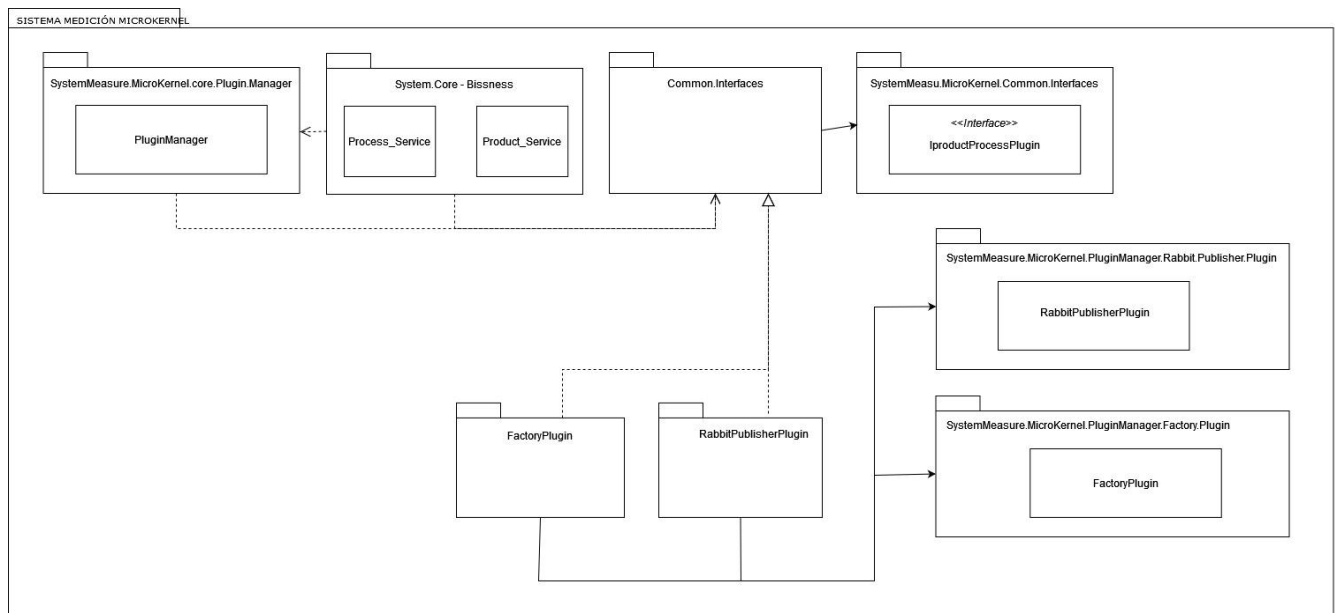
Cod. Escenario: EAC05	
<b>Descripción</b>  <b>El sistema debe contar con una interfaz intuitiva, sencilla y que permita realizar todos los procedimientos de manera rápida.</b>	
Interesado: Usuario	Atributo: <b>Usabilidad</b>
<b>Validación del Escenario</b>	
<b>Origen del estímulo</b>	Una interfaz fácil de entender e interactuar para el usuario.
<b>Estímulo</b>	Uso
<b>Entorno</b>	En ejecución
<b>Artefacto</b>	Aplicación
<b>Respuesta</b>	Un entorno amigable y estimulante para el usuario.
<b>Medida de respuesta</b>	Automático

## DIAGRAMA MODELO CASOS DE USO

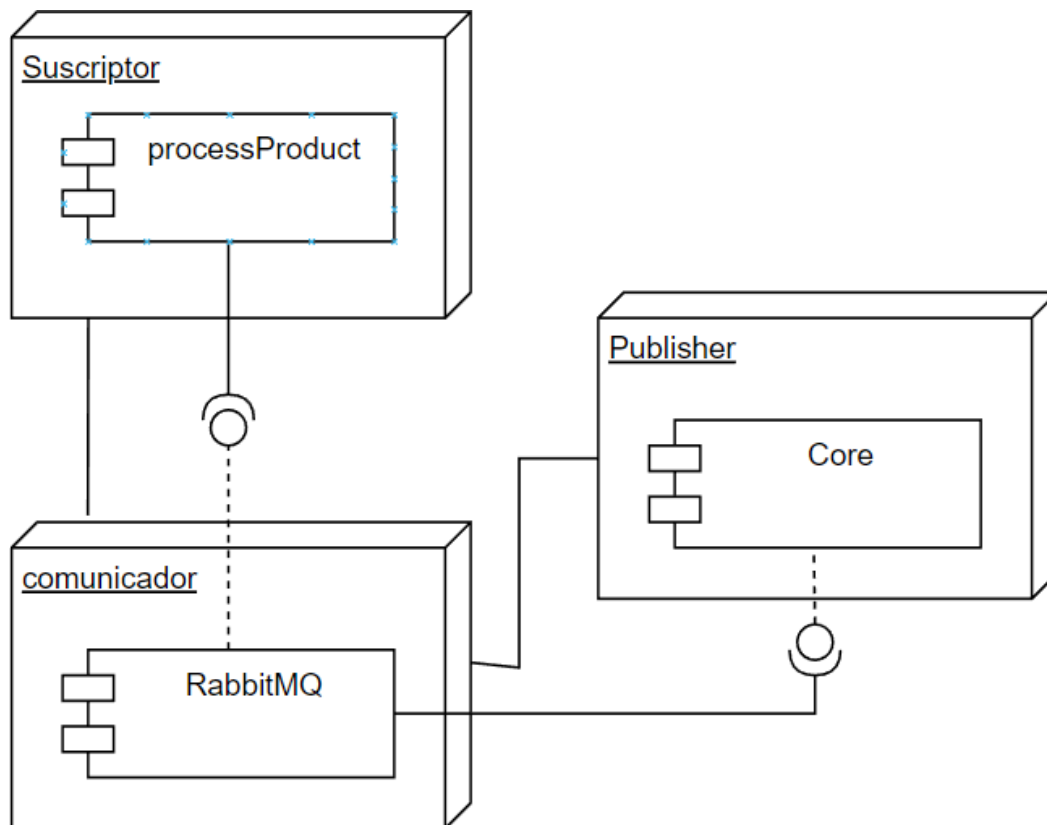


## Módulos (UML)



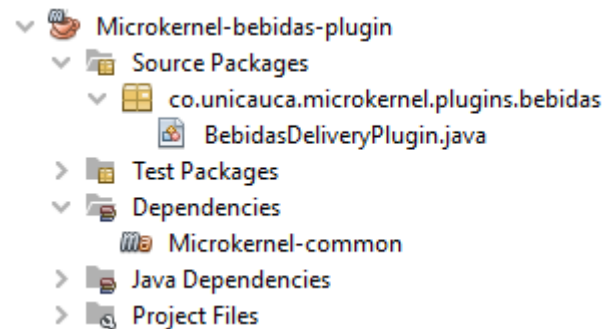
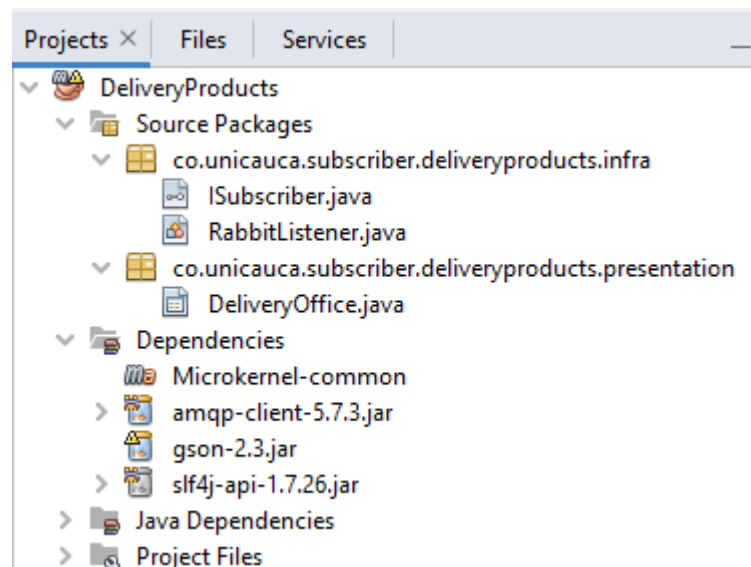
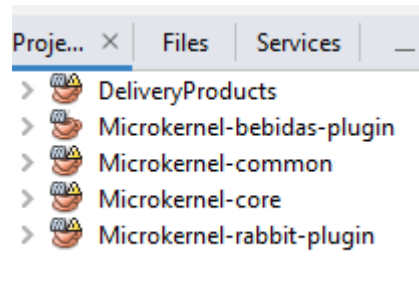


## Componentes y conectores (UML)



## Vista de Implementación

### Implementación (Maven, UML)



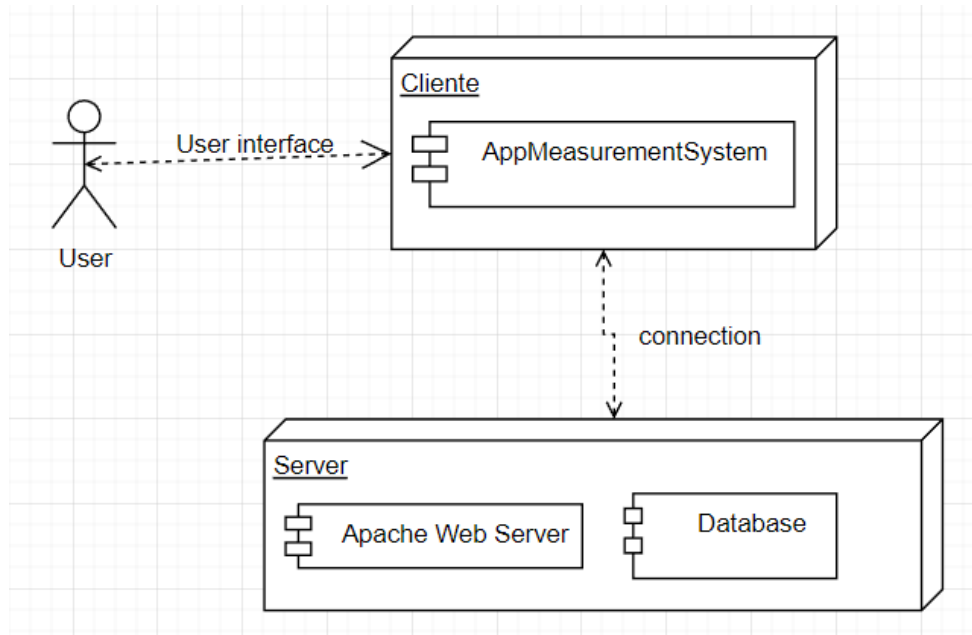
- Microkernel-common
  - Source Packages
    - co.unicauca.microkernel.common.entities
      - Medicion.java
      - Product.java
    - co.unicauca.microkernel.common.interfaces
      - IMedicionPlugin.java
      - IPublisherPlugin.java
  - Test Packages
  - Dependencies
  - Java Dependencies
  - Project Files

- Microkernel-core
  - Source Packages
    - co.unicauca.microkernel.app
      - DeliveryServiceApplication.java
    - co.unicauca.microkernel.business
      - ProductService.java
      - ServicioMedicion.java
    - co.unicauca.microkernel.infra
      - Publisher.java
    - co.unicauca.microkernel.plugin.manager
      - PluginManager.java
    - co.unicauca.microkernel.presentation
      - GUISendProduct.java
  - Test Packages
  - Other Sources
  - Dependencies
    - AbsoluteLayout-SNAPSHOT.jar
    - Microkernel-common
    - gson-2.8.6.jar
  - Runtime Dependencies
  - Java Dependencies
  - Project Files

- Microkernel-rabbit-plugin
  - Source Packages
    - co.unicauca.microkernel.plugins.rabbit
      - RabbitPublisherPlugin.java
  - Dependencies
    - Microkernel-common
    - amqp-client-5.7.3.jar
    - slf4j-api-1.7.26.jar
  - Java Dependencies
  - Project Files

## Vista de Instalación

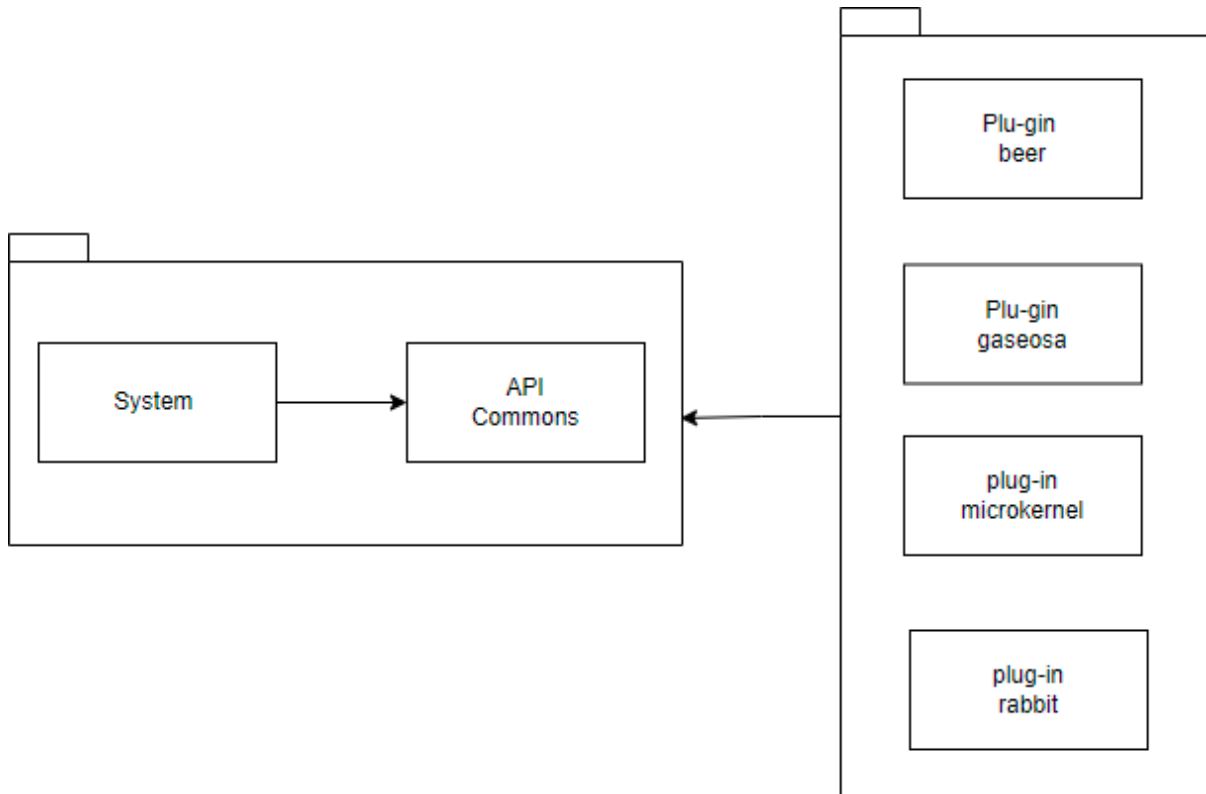
### Instalación (Allocation, UML)





## Vista de despliegue

### Vista de despliegue (UML)



## Decisiones de arquitectura y justificación

**Tipo de aplicación:** Se tomó la decisión de desarrollar una aplicación que facilite a futuros desarrolladores implementar y modificar un sistema de medición de una industria de producción, esta aplicación es tipo framework y permite hacerle un control de calidad de los productos de una línea de fabricación y modificar de acuerdo a los sensores y datos necesarios.

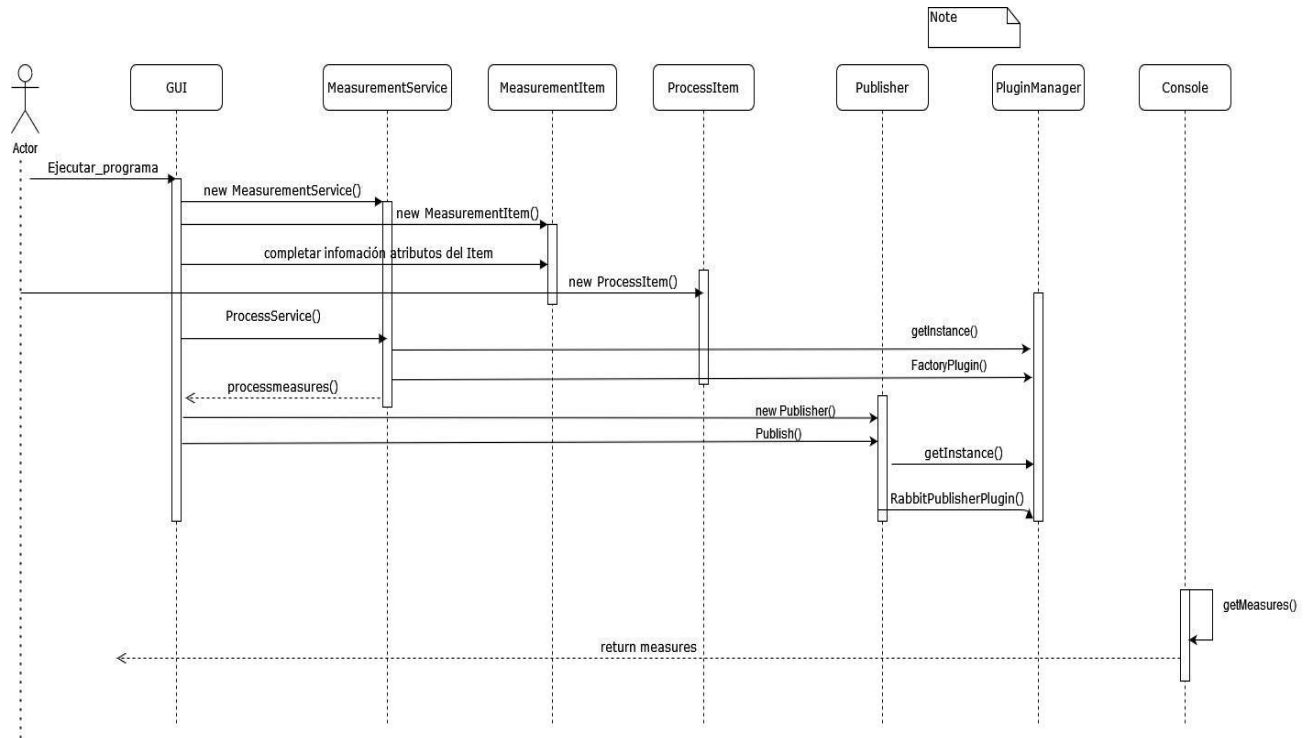
**Estilo arquitectónico:** se optó por el estilo arquitectónico publicador-subscriptor junto con microkernel debido a la conveniencia para el proyecto, es un modelo sencillo y escalable que facilita la división en las tareas de los distintos módulos del sistema.

### Uso de tecnologías:

- Se decidió utilizar la tecnología java sockets la cual consiste en puntos finales de enlaces de comunicaciones entre procesos. Los procesos los tratan como descriptores de ficheros, de forma que se pueden intercambiar datos con otros procesos transmitiendo y recibiendo a través de sockets.

## Diseño detallado de módulos implementados

### Diagramas de secuencia:



## Capturas de pantallas de espacio de trabajo SCRUM

- Backlog

3/8/22, 20:01 ProyectoUnoSoftware: tablero ágil - Jira

Tu trabajo ▾ Proyectos ▾ Filtros ▾ Paneles ▾ Personas

Proyectos / ProyectoUnoSoftware

### Backlog

Epic ▾ Insights

- PROYEC-35 PIPAC-16 ver el tipo de arquitectura para la entrega final y hacer un reporte
- PROYEC-36 PIPAC-16 reunion de seguimiento de tareas
- PROYEC-22 PIPAC-3 como usuario deseo poder ingresar el nombre del producto
- PROYEC-23 PIPAC-4 como usuario deseo poder ingresar el peso del producto
- PROYEC-24 PIPAC-5 convertir el peso en mg
- PROYEC-25 PIPAC-6 el publiher-subscriber debe poder enviar el mensaje que recibio
- PROYEC-26 PIPAC-7 Hacer el diagrama de secuencia del proyecto
- PROYEC-27 PIPAC-8 hacer el diagrama de caso de uso
- PROYEC-28 PIPAC-9 hacer el diagrama de modulos
- PROYEC-29 PIPAC-10 hacer el diagrama de vista de componentes y conectores
- PROYEC-30 PIPAC-11 hace la vista de despliegue
- PROYEC-31 PIPAC-12 hacer el diagrama de clases
- PROYEC-32 PIPAC-13 subir documentos al repositorio
- PROYEC-34 PIPAC-15 acomodar el codigo para el sprint 2
- PROYEC-37 PIPAC-17 Organizar documentacion del sprint 2 y completarla

+ Crear incidencia

▼ Backlog (2 incidencias) 0 0 0 Crear sprint

- PROYEC-38 PIPAC-18 Reunión final
- PROYEC-39 PIPAC-19 Revisión de documento y cogido final

<https://acuatindioy.atlassian.net/jira/software/projects/PROYEC/boards/1/backlog> 1/1

- **Tablero**

3/8/22, 20:10 Tablero PROYEC: tablero ágil - Jira

Tu trabajo **Proyectos** Filtros Paneles Personas Aplicaciones

Proyectos / ProyectoUnoSoftware

### Tablero Sprint 2

framework para facilitar la reutilización en la construcción de sistemas de medición. Usando arquitectura de Microkernel y publisher-subscriber

Restantes: 0 días Terminar sprint

AGRUPO POR Nada Insights

POR HACER 1 INCIDENCIA	EN CURSO 3 INCIDENCIAS	LISTO 15 INCIDENCIAS ✓
<div>PIPAC-19 Revisión de documento y código final</div> <div>PROYEC-39</div>	<div>PIPAC-15 acomodar el código para el sprint 2</div> <div>PROYEC-34</div> <div>PIPAC-18 Reunión final</div> <div>PROYEC-38</div> <div>PIPAC-17 Organizar documentación del sprint 2 y completarla</div> <div>PROYEC-37</div>	<div>PIPAC-1 clonar el repositorio</div> <div>PROYEC-20 ✓</div> <div>PIPAC-2 como usuario deseo poder ingresar el id del producto</div> <div>PROYEC-21 ✓</div> <div>PIPAC-16 ver el tipo de arquitectura para la entrega final y hacer un reporte</div> <div>PROYEC-35 ✓</div> <div>PIPAC-16 reunión de seguimiento de tareas</div> <div>PROYEC-36 ✓</div>

<https://acuatindioy.atlassian.net/jira/software/projects/PROYEC/boards/1>

1/1

**URL repositorio con código fuente:**

<https://github.com/juanoa26/Sistema-de-medicion>

**URL jira:**

<https://acuatindioy.atlassian.net/jira/software/projects/PROYEC/boards/1/backlog>