

Pontificia Universidad  
**JAVERIANA**  
Colombia

## **Pontificia Universidad Javeriana**

Departamento de Ingeniería de Sistemas

### **Sistemas Distribuidos**

### **Entrega Uno Corregida Proyecto**

### **Nombre de integrantes:**

Santiago Camilo Rey Benavides  
Lara Salcedo Franco

Profesor: Rafael Vicente Páez Mendez

Noviembre 23, 2023  
Bogotá, Colombia

## Tabla de contenido

1. DISEÑO DEL SISTEMA .....	2
1.1. DIAGRAMA DE COMPONENTES .....	2
1.1. DIAGRAMA DE CLASES .....	3
1.2. DIAGRAMA DE SECUENCIA .....	3
2. MODELOS ARQUITECTURALES Y FUNDAMENTALES .....	4
2.1. MODELO DE INTERACCIÓN .....	4
2.2. MODELO DE FALLOS .....	5
2.3. MODELO DE SEGURIDAD .....	6
3. PROTOCOLO DE PRUEBAS .....	6
3.1. ESPECIFICACIONES DE HARDWARE Y SOFTWARE .....	6
3.2. PRUEBAS DE RENDIMIENTO (DE CARGA).....	7
3.3. PRUEBAS DE TOLERANCIA A FALLOS .....	8
3.4. PRUEBAS CON UN PATRÓN DE COMUNICACIÓN DIFERENTE .....	9

## 1. DISEÑO DEL SISTEMA

### 1.1. DIAGRAMA DE COMPONENTES

Aquí se puede observar cómo van a estar distribuidos entre máquinas virtuales ofrecidas por la universidad y por nuestros computadores personales conectados a la VPN de esta misma.

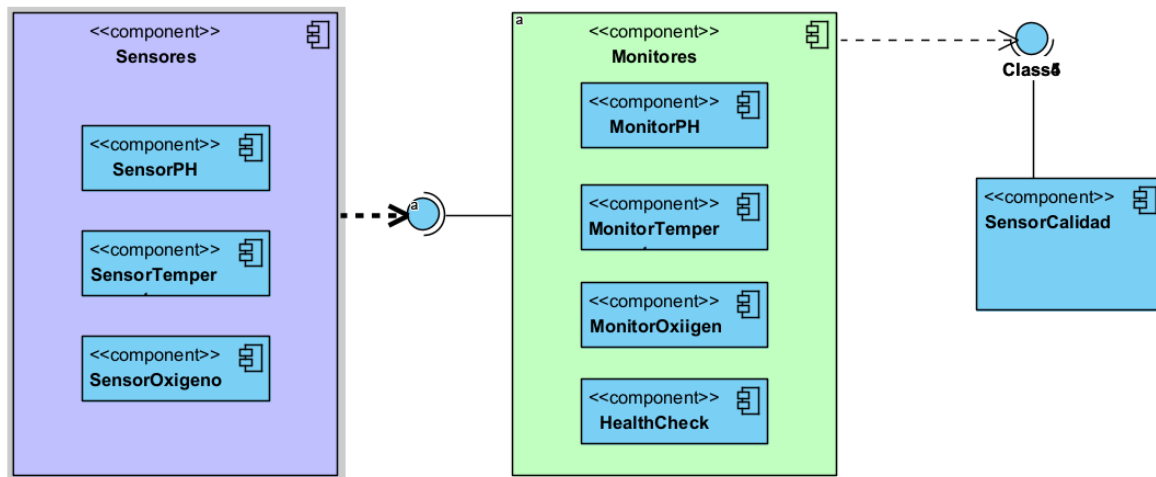
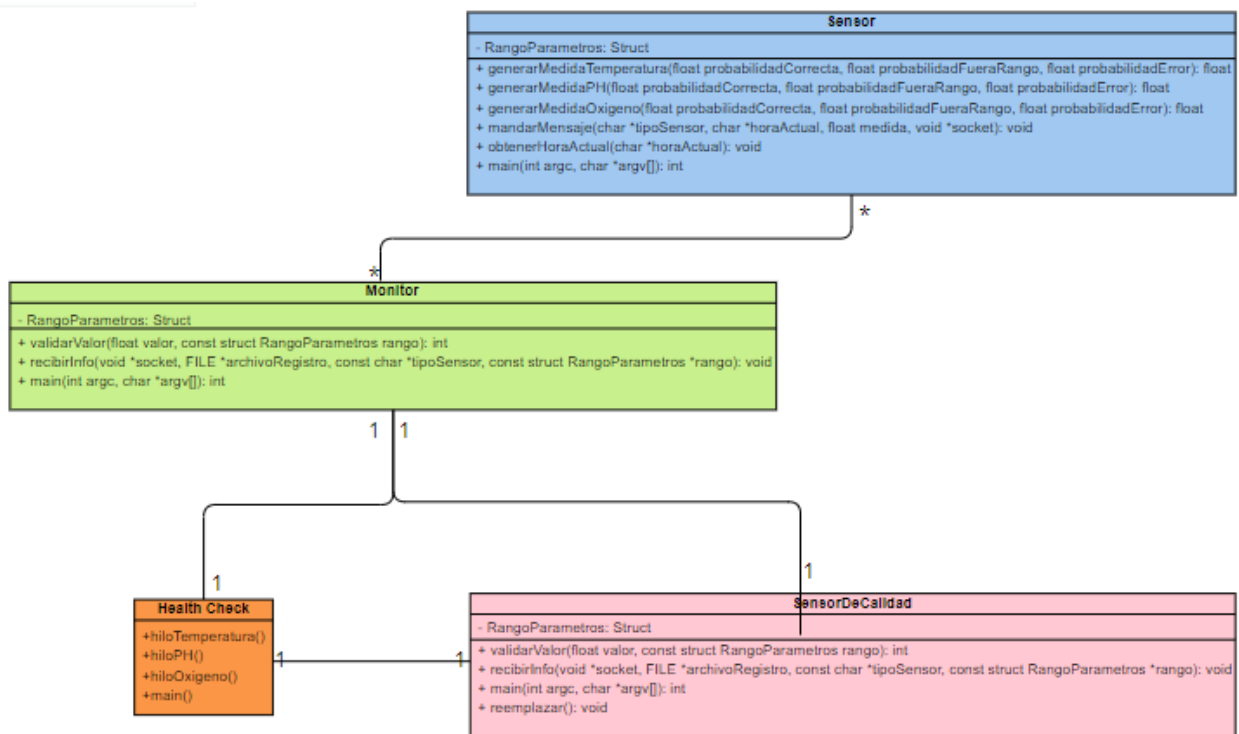


Ilustración 1 Diagrama de componentes

## 1.1. DIAGRAMA DE CLASES

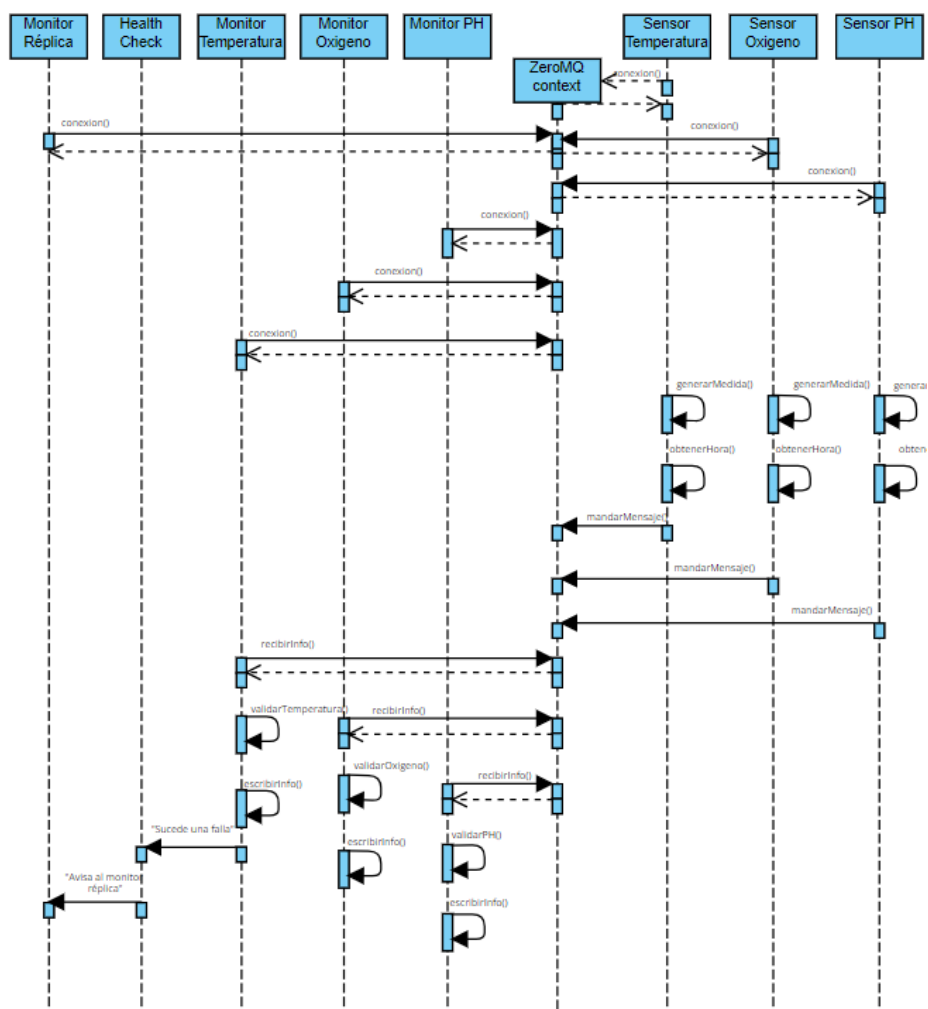


## 1.2. DIAGRAMA DE SECUENCIA

En este diagrama de Secuencia se puede observar el funcionamiento general del sistema, como se puede ver al inicio de todo en un funcionamiento ideal los monitores y los sensores crean y se conectan al contexto de ZeroMQ (Esto sucedería todo al tiempo, pero es difícil de representar en el diagrama). Posteriormente los sensores generan las

medidas y obtienen la información de la hora para después mandar toda la información al contexto de ZeroMQ donde esta se guarda dependiendo del tipo de información que es, ya en el momento en el que la información es enviada por los sensores los monitores tendrán la capacidad de leer esa información para posteriormente validarla y escribirla en su archivo de texto correspondiente.

En caso de que algún monitor tenga alguna falla la clase Health Check estará activa y recibirá esta información y posteriormente notificará al monitor réplica para que este reemplace al monitor que falló.



## 2. MODELOS ARQUITECTURALES Y FUNDAMENTALES

### 2.1. MODELO DE INTERACCIÓN

- **Sensores:** Se configurarán tres tipos de sensores: PH, temperatura y oxígeno, representados por procesos simulados. Cada proceso sensor recibirá tres argumentos: el tipo de sensor (PH, temperatura u oxígeno), la frecuencia de envío de mediciones y un archivo de configuración común para los tres tipos. En este archivo se especificarán tres probabilidades que determinarán la calidad de las mediciones: probabilidad de valores correctos, probabilidad de valores fuera del rango y probabilidad de errores (en este proyecto, representados por valores negativos). Por ejemplo, si las probabilidades son 0.6, 0.3 y 0.1 respectivamente, de 10 mediciones de un sensor de temperatura, el 60% será correcto, el 30% estará fuera del rango y el 10% será inválido.
- **Monitores:** Estos procesos se suscribirán a un tipo específico de sensor (PH, temperatura u oxígeno). Recibirán mediciones correspondientes a estos sensores, validarán la integridad de los datos y su rango. Almacenarán todas las mediciones sin errores, incluso si están fuera del rango. Cada vez que se detecte una medición fuera del rango, se generará una alerta al Sistema de Calidad.
- **Sistema de Calidad (SC):** Su función principal será recibir las alertas de los monitores y desplegarlas en pantalla.
- **HealthCheck:** Se implementará un componente de HealthCheck para monitorear la funcionalidad de los monitores. Este proceso verificará constantemente si todos los monitores están activos y funcionando correctamente. En caso de detectar la inactividad de uno o más monitores, el HealthCheck asumirá la tarea de dicho monitor inactivo, reasignando automáticamente su carga de trabajo. El reemplazo se realizará sin interrupciones en las operaciones, asegurando la continuidad del servicio.

## 2.2. MODELO DE FALLOS

Para abordar la tolerancia a fallos, se implementó un modelo integral de comunicación basado en el patrón publicador/suscriptor utilizando ZeroMQ entre todos los sistemas. Se introdujo una conexión de tipo publicador en el monitor y se creó un componente adicional, el "Sistema de Calidad", que se conecta como suscriptor en un puerto separado al utilizado anteriormente. Además, se diseñó un sistema de "Health Check" que monitorea continuamente la conexión con el Monitor. En caso de interrupción, el Health Check activa un protocolo para transferir funciones al Sistema de Calidad y registra la información pertinente en un archivo hasta que se restablezca la conexión con el Monitor.

### 2.3. MODELO DE SEGURIDAD

Se propone implementar un cifrado de extremo a extremo en la comunicación entre los diferentes componentes del sistema, lo que garantiza la confidencialidad de los datos transmitidos. Además, se considerará la autenticación de los dispositivos conectados, validando su identidad antes de permitir la transmisión o recepción de información.

## 3. PROTOCOLO DE PRUEBAS

### 3.1. ESPECIFICACIONES DE HARDWARE Y SOFTWARE

Las máquinas que vamos a utilizar para los procesos de los sensores y los monitores son tres principales:

Un computador personal de un integrante del grupo con Windows 11 y con las siguientes especificaciones:

Procesador	11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz 2.50 GHz
RAM instalada	8,00 GB (7,78 GB usable)
Identificador de dispositivo	D7E87AAD-3187-442B-AB42-98860F17980E
Id. del producto	00342-43271-76497-AAOEM
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla

Para poder usar correctamente este computador necesitaremos conectarlo a la red de la universidad a través de la VPN de F5

Además, vamos a usar varias máquinas virtuales proporcionadas por la universidad a los miembros del equipo.

Estas máquinas tienen las siguientes características:

Device name	VINGC-GEN283
Processor	Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz 2.19 GHz (3 processors)
Installed RAM	16,0 GB
Device ID	9116EE25-980F-4467-894E-D46FA3DDF350
Product ID	00454-10000-00001-AA371
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Y una versión de Windows 10 enfocada para servidores:

OS Name	Microsoft Windows Server 2022 Standard
Version	10.0.20348 Build 20348

### 3.2. PRUEBAS DE RENDIMIENTO (DE CARGA)

#### Objetivo:

Evaluar el comportamiento del sistema bajo diferentes cargas de sensores, monitores y comunicaciones, midiendo tiempos de ejecución y capacidad de respuesta del sistema.

#### Escenarios de Prueba:

##### Variación en la Cantidad de Sensores:

- Ejecutar el script con 10, 100 y 1000 sensores simulados.
- Cronometrar el tiempo de respuesta para la publicación de las mediciones y su procesamiento por los monitores.

##### Comportamiento del Sistema:

- Observar cómo varía el tiempo de respuesta a medida que se incrementa el número de sensores.
- Registrar la diferencia en los tiempos de procesamiento del sistema al aumentar la carga.

##### Estabilidad del Sistema:

- Comprobar la estabilidad y continuidad de las operaciones al incrementar la cantidad de sensores.
- Medir la capacidad de los monitores y el sistema de calidad para manejar mayores cargas.

##### Parámetros de Medición:

- **Tiempo de Respuesta:** Diferencia entre el momento de envío y recepción de las mediciones.
- **Utilización del Recurso del Sistema:** Observar el uso del procesador, la memoria y otros recursos relevantes durante la ejecución de las pruebas.
- **Capacidad de Tolerancia a Fallos:** Evaluar cómo responde el sistema ante la pérdida de un monitor o la activación del HealthCheck.

##### Métodos de Recopilación de Datos:

- Utilización del script de Bash para ejecutar los diferentes escenarios de prueba.

- Registro de tiempos de ejecución, uso de recursos y posibles fallos durante las pruebas.

#### **Resultados Esperados:**

- Identificación de la capacidad máxima de procesamiento del sistema.
- Análisis de la estabilidad y capacidad de respuesta ante diferentes cargas.

### **3.3. PRUEBAS DE TOLERANCIA A FALLOS**

#### **Objetivo:**

- Evaluar la capacidad del sistema para manejar situaciones de fallo o caída de componentes, y cómo se recupera el sistema para mantener la continuidad de las operaciones.

#### **Escenarios de Prueba:**

##### **Simulación de Fallo de Monitores:**

- Detener intencionalmente uno o más procesos monitores mientras el sistema está en funcionamiento.
- Observar cómo responde el sistema y si los procesos réplica asumen la carga de manera transparente.

##### **HealthCheck y Cambio Automático de Carga:**

- Detener un monitor y observar cómo el proceso HealthCheck detecta la falla.
- Verificar la reconexión automática del proceso réplica y cómo asume la carga de trabajo.

##### **Continuidad de Operaciones:**

- Evaluar si las operaciones continúan sin interrupciones después de la falla y cambio a un proceso réplica.
- Comprobar si las mediciones y alarmas se siguen gestionando correctamente.

##### **Parámetros de Medición:**

- Tiempo de Recuperación: Tiempo requerido para que el proceso réplica asuma la carga.
- Continuidad de Operaciones: Verificar si las mediciones y las alertas se procesan sin pérdida de datos o interrupciones significativas.
- Funcionamiento del HealthCheck: Observar la efectividad del HealthCheck para detectar y reaccionar ante fallas.

##### **Métodos de Recopilación de Datos:**

- Monitoreo manual durante las pruebas para registrar el tiempo de recuperación y la continuidad de las operaciones.



- Registro de las acciones realizadas y los resultados obtenidos durante las simulaciones de fallos.

#### **Resultados Esperados:**

- Confirmación de la capacidad del sistema para detectar y responder a fallos de manera automática.
- Verificación de la continuidad de las operaciones y la rápida recuperación mediante procesos réplica.
- Validación del correcto funcionamiento del HealthCheck para la detección temprana de fallos y activación de procesos réplica.

### **3.4. PRUEBAS CON UN PATRÓN DE COMUNICACIÓN DIFERENTE**

#### **Objetivo:**

Evaluar el impacto del cambio del patrón de comunicación en el sistema, comparándolo con el patrón original utilizado (publicador/suscriptor).

#### **Escenarios de Prueba:**

##### **Implementación de un Nuevo Patrón de Comunicación:**

- Cambiar el patrón de comunicación utilizado en el sistema, reemplazando el patrón publicador/suscriptor por otro, como el modelo push/pull o req/rep de ZeroMQ.
- Comparar la eficacia y eficiencia del nuevo patrón de comunicación con el modelo original.

##### **Carga de Trabajo Incrementada:**

- Aumentar la carga del sistema (número de sensores o frecuencia de mediciones) mientras se utiliza el nuevo patrón de comunicación.
- Medir el impacto en el rendimiento y la capacidad del sistema para manejar la carga adicional.

##### **Análisis de Rendimiento:**

- Comparar el rendimiento del sistema (tiempo de procesamiento, latencia, utilización de recursos) utilizando el nuevo patrón de comunicación frente al patrón original.
- Evaluar si el nuevo patrón proporciona ventajas significativas en términos de rendimiento o eficiencia.

##### **Parámetros de Medición:**

- Tiempo de Procesamiento: Comparar los tiempos de procesamiento de las mediciones y alarmas con el nuevo patrón y el modelo original.
- Latencia: Medir el tiempo que tarda un mensaje desde su envío hasta su recepción con ambos patrones.

- Utilización de Recursos: Observar el uso de recursos del sistema (CPU, memoria) con cada patrón bajo diferentes cargas de trabajo.

**Métodos de Recopilación de Datos:**

- Instrumentación del sistema para registrar tiempos de procesamiento y latencia.
- Herramientas de monitoreo para medir la utilización de recursos del sistema durante las pruebas.

**Resultados Esperados:**

- Identificación de las ventajas y desventajas del nuevo patrón de comunicación en comparación con el modelo original.
- Determinación de si el nuevo patrón mejora el rendimiento del sistema en términos de latencia, tiempo de procesamiento o utilización de recursos.
- Recomendaciones sobre la viabilidad y utilidad del nuevo patrón de comunicación en el contexto específico del sistema.