 Sistema de Medición de Calidad del Agua

**PONTIFICIA UNIVERSIDAD JAVERIANA**

**FACULTAD DE INGENIERIA**

**DEPARTAMENTO INGENIERIA DE SISTEMAS**

**Proyecto Introducción a Sistemas Distribuidos**

**Período Académico 2023-03**

# Objetivos del Proyecto

* Desarrollar una solución a un problema de estructura distribuida
* Utilizar patrones de comunicación síncronos y asíncronos. Utilizar el patrón publicador/suscriptor.
* Resolver problemas presentes en sistema distribuidos, tales como fallas en los componentes.
* Reconocer elementos de desempeño y calidad asociados a la implementación de un sistema distribuido



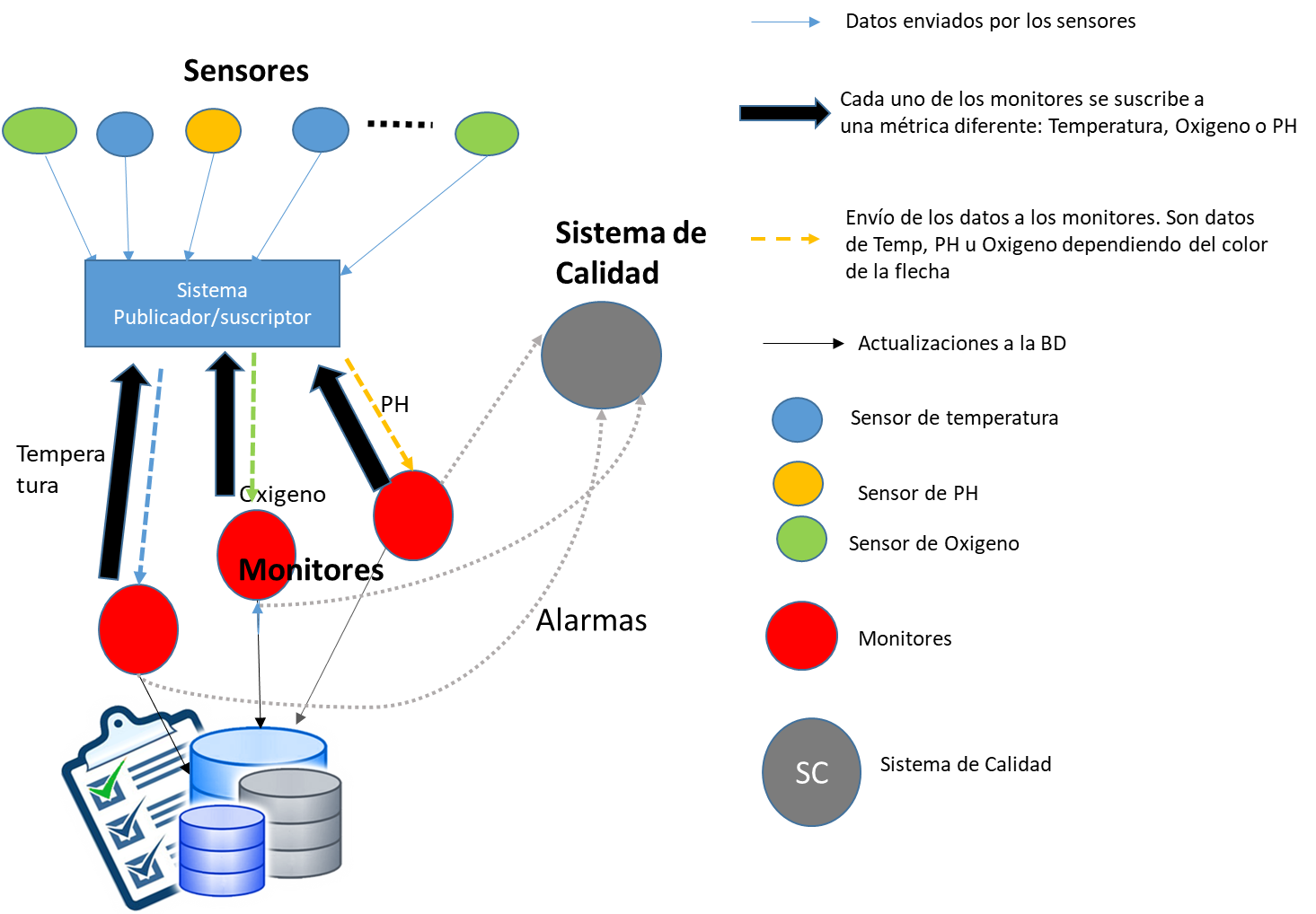
# Contexto

El agua es uno de los recursos vitales en nuestro planeta. Con una población en aumento, es importante monitorear la calidad de este recurso tan importante para detectar cambios y tomar las acciones requeridas. La calidad de agua es una medida de qué tan adecuada es una fuente hídrica para un uso específico, por ejemplo, agricultura o generación de energía. Hay varios parámetros esenciales que nos ayudan a medir la calidad del agua, entre ellos el PH, la temperatura, la conductividad, el oxígeno disuelto y la turbidez (<https://www.hannacolombia.com/blog/post/190/guia-para-la-medicion-en-campo-calidad-del-agua)>

En este proyecto realizaremos un sistema distribuido sencillo y tolerante a fallas, donde se simula la medición de tres parámetros de una reserva de agua: PH, temperatura y oxígeno disuelto. Estos parámetros se enviarán a procesos monitores, que los almacenarán en forma adecuada y avisarán al sistema de calidad en caso de que se genere alguna alerta con los indicadores medidos.

# Descripción del Sistema a Desarrollar

A continuación, se explica cada uno de los componentes de la arquitectura de software (figura 1)



## Sensores

Existirán tres tipos de sensores: **de PH, de temperatura y de oxígeno** y serán simulados por procesos. Al levantar cada proceso sensor, éste deberá recibir 3 argumentos: **tipo de sensor (temperatura, PH u oxigeno), cada cuanto tiempo (t) se va a enviar la medición a los procesos monitores y un archivo de configuración**. En la tabla 1 se pueden observar valores válidos para cada uno de los parámetros de calidad. En el archivo de configuración (único para los tres tipos de procesos) se encontrarán tres probabilidades: probabilidad de producir valores dentro del rango, probabilidad de producir valores fuera del rango y probabilidad de producir valores erróneos (para el caso del proyecto serán valores negativos). Por ejemplo, si el archivo de configuración contiene las siguientes probabilidades:

0.6 Valores correctos

0.3 Valores fuera del rango

0.1 Errores

De 10 medidas producidas por el sensor de temperatura, 6 (60%) de ellas serán correctas, 3 (30%) estarán fuera del rango y 1 (10%) será invalida, por ejemplo:

*68 69 –1 90 75 81 70 100 80 95*

**Tabla 1: Rangos de los parámetros de calidad**

|  |  |  |
| --- | --- | --- |
| Parámetro | **Valor mínimo** | **Valor máximo** |
| **Temperatura** | 68°F | 89°F |
| **PH** | 6.0 | 8.0 |
| **Oxígeno Disuelto** | 2 Mg/L | 11Mg/L |

Las medidas del sensor serán números aleatorios que atiendan las probabilidades descritas en el archivo de configuración. Dichas medidas se enviarán o publicarán (patrón publicador/suscriptor) cada **t** unidades de tiempo. Cada medición debe ir acompañada del tiempo (u hora) en el cuál se produjo.

**Nota:** Recuerde que pueden ejecutarse varios sensores al tiempo. Debería poder levantarse más de un sensor de cada tipo.

## Monitores

Son procesos que se suscriben a un determinado tipo de sensor. Reciben las medidas de los sensores correspondientes, validan que los datos no contengan errores y se encuentren en rangos razonables y almacenan todas las medidas sin errores, aunque estén fuera del rango. Cada vez que encuentran una medida fuera del rango deben generar una alarma al **sistema de calidad**. **Un proceso monitor puede suscribirse solamente de un tipo de indicador Temperatura, PH u oxigeno.** Este indicador se recibirá, al iniciar el programa, como argumento del programa principal.

## Sistema de Calidad (SC)

La única función de este proceso es esperar las posibles alarmas de los monitores e imprimir las alertas por pantalla.

# Detalles de Implementación

**Comunicación:** la comunicación entre los sensores y monitores se debe hacer con el modelo publicador/suscriptor. La comunicación entre los monitores y el sistema de calidad se puede realizar utilizando cualquiera de los otros patrones que ofrezca la librería ZeroMQ y que el equipo considere adecuado para la notificación.

**Fallas:** debe implementar una arquitectura tolerante a fallas, considerando **una posible falla del componente monitor.** Una forma de implementación es tener un componente réplica que realice las mismas funciones. Debe existir también otro proceso encargado de chequear que todos los monitores están o no en funcionamiento (health check). Cuando se detecte que uno de estos procesos deja de funcionar, el proceso réplica debe asumir la carga. La reconexión del nuevo proceso y distribución de cargas debe hacerse de forma automática, es decir, será transparente para el cliente; las operaciones deben continuar en el punto en que se habían dejado y no iniciar desde el principio. En el informe final debe indicar que patrón de tolerancia a fallas utilizó.

**Necesario para la evaluación**: el día de la sustentación es importante que se pueda observar:

* Operaciones que van realizando cada uno de los procesos y resultado de la operación, si aplica.
* Información guardada en la BD (o en un archivo)
* El funcionamiento del proyecto distribuyendo los componentes en al menos tres computadoras (o máquinas virtuales): sensores en un computador, uno o dos procesos monitores en otro computador, monitor o monitores restantes y el sistema de calidad en otro computador.

# Rendimiento del Sistema

## Variables para medir el rendimiento del sistema

1. Defina variables que permitan medir el rendimiento de su sistema, por ejemplo: tiempo desde que se produce hasta que se almacena la medición , tiempo en que tarda la llegada de cada alarma al sistema de calidad, utilización de los procesadores, etc. **Debe definir al menos dos variables.** Suponga que los relojes de las distintas máquinas están sincronizados.
2. Establezca el procedimiento para tomar el valor de dichas variables, por ejemplo, va a instrumentar el programa, va a usar ciertas herramientas especiales de monitoreo, ¿cuáles serán?, etc.

## Elementos que afectan el rendimiento del sistema

Existen muchos elementos que afectan el desempeño del sistema, por ejemplo: la carga, los patrones de comunicación utilizados, la congestión en la red o en los servidores, etc.

La siguiente actividad que Ud. va a realizar es incorporar un elemento a su sistema que afecte su desempeño. Llamaremos **sistema A** al sistema sin este elemento (el sistema inicial que implementó) y **sistema B,** aquel que tienen un elemento que afecta su desempeño. A continuación, les damos ejemplos de factores que pueden afectar el rendimiento:

* **Aumentar la carga**: disminuir el tiempo de generación de medidas desde los sensores o aumentar el número de procesos sensores.
* **Patrones de comunicación**: en lugar del patrón publicador-suscriptor usar otro patrón como el push/pull de la librería (<https://programmerclick.com/article/915097237/>) para hacer llegar las mediciones de los sensores a los monitores.
* **Congestión en los servidores**: genere trabajos dummy adicionales en los servidores donde se ejecutan los monitores.

# Comparación de los sistemas

Una vez que el grupo decida las variables a medir, cómo las va a medir y qué elemento va a incorporar para afectar el rendimiento del sistema, realice medidas (experimentación) para comparar el sistema A con el B. Coloque los resultados de las medidas en gráficos y analice los resultados obtenidos. Exponga las conclusiones.

# Herramientas que puede utilizar

A continuación sugerimos algunas herramientas:

### Para generar Carga:

Apache Jmeter,  Grafana k6

### Para obtener el valor de las métricas de rendimiento

* Comandos del sistema operativo
* Scalene (<https://github.com/plasma-umass/scalene>)
* Visualvm (<https://visualvm.github.io/>)
* UBER JVM PROFILER (https://www.uber.com/en-CO/blog/jvm-profiler/)
* Pyroscope (<https://pyroscope.io/>)
* Jensor (<https://jensor.sourceforge.net/>)

# PRIMERA ENTREGA

La primera entrega se realizará en la semana 10 a través de la plataforma. Las sustentaciones serán el día correspondiente a la asignatura de acuerdo con el grupo.

La primera entrega consta de un informe donde se debe especificar:

* Diseño del sistema: Diagrama de clases y diagrama de secuencia. Este diseño debe incluir el o los componentes para enmascarar las fallas del sistema.
* El protocolo de pruebas que utilizará para la entrega final (considere **todos** los tipos de prueba que deben realizarse a un sistema)
* Describa cómo va a afectar el rendimiento de su sistema para la entrega final y cómo va a comparar el rendimiento de los sistemas A y B: variables a medir,herramientas de medición, plataforma de hw y sw usadas, método para generar la carga y herramientas de medición.
* Implementación inicial: deben tener implementada al menos la generación de mediciones desde los sensores y su recepción por parte de los procesos monitores. Esto debe estar funcionando en dos computadoras.

Se deben implementar los requerimientos de acuerdo con el enunciado y la rúbrica.

## El día de la sustentación, cada equipo tendrá 10 minutos para mostrar sus resultados y responder las preguntas que tengan los profesores.

# SEGUNDA ENTREGA

La entrega se realizará en la semana 17 a través de la plataforma. Las sustentaciones se harán los días que correspondan para cada curso. El día de la sustentación los integrantes del equipo deben **mostrar la funcionalidad del proyecto**. Deben estar presentes todos los integrantes del grupo.

La entrega se compone de:

* En un archivo .zip código fuente de los programas que conforman el sistema y un archivo Readme donde indique cómo ejecutarlo.
* Se debe complementar la documentación de la primera entrega
* Un video de máximo 10 minutos donde muestra la topología implementada y explicar los siguientes aspectos de su proyecto:
  1. Distribución de componentes en máquinas.
  2. Librerías y patrones usados
  3. Cómo se implementó la arquitectura confiable (tolerante a fallas)
* Un informe de máximo 4 páginas donde explican: experimentos realizados (especificaciones de hw y sw donde se realizaron las medidas, variables e instrumentos de medición), resultados obtenidos (si es posible acompañados por gráficos o tablas), conclusiones.

Equipos de Trabajo.

El proyecto se realizará en grupos de trabajo de máximo tres personas

**No puede existir replicación de documentos ni de código fuente entre grupos, lo cual se consideraría plagio.**

# Calificación I Entrega

**20 ptos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indicador | **Valoración en puntos /20** | **Excelente** | **Competente** | **Deficiente** |
| Informe | 2 | 2 | 1 | <1 |
| Diseño del Proyecto | 5 | [5,4] | (4, 3] | < 3 |
| Protocolo de pruebas | 3 | [3,2.5] | (2.5,1] | < 1 |
| Modelos del sistema | 2 | 2 | 1 | <1 |
| Evaluación de Rendimiento | 3 | [3,2.5] | (2.5,1] | < 1 |
| Implementación Inicial | 5 | [5,4] | (4, 3] | < 3 |
| total | 20 |  |  |  |

# Calificación II Entrega

**20 ptos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Indicador | **Valoración en puntos /30** | **Excelente** | **Competente** | **Deficiente** |
| Implementación de los Sensores | 2 | [2,1.5] | (1.5,1] | < 1 |
| Monitores | 3 | [3,2.5] | (2.5,1] | < 1 |
| Sistema de Calidad | 2 | 2 | 1 | <1 |
| Corrida en 3 máquinas | 2 | 2 | 1 | < 1 |
| Tratamiento de Fallas del monitor | 2 | 2 | 1 | < 1 |
| Código | 2 | 2 | 1 | <1 |
| Protocolo de pruebas funcionales | 2 | [2,1.5] | (1.5,1] | < 1 |
| Medidas de Rendimiento, Reporte y Análisis de Resultados obtenidos | 2 | [2,1.5] | (1.5,1] | < 1 |
| Sustentación, video | 3 | [3,2] | (2, 1] | < 1 |
| Total | 20pts |  |  |  |