



**PONTIFICIA UNIVERSIDAD JAVERIANA**  
**FACULTAD DE INGENIERIA**  
**DEPARTAMENTO INGENIERIA DE SISTEMAS**  
Proyecto Introducción a Sistemas Distribuidos

Período Académico 2024-10

# Sistema Distribuido: Cloud, Fog, Edge

## Objetivos del Proyecto

- Desarrollar una solución a un problema de estructura distribuida
- Utilizar patrones de comunicación síncronos y asíncronos.
- Resolver problemas presentes en sistemas distribuidos, tales como fallas en los componentes y persistencia de datos.
- Reconocer atributos de calidad (e.g. desempeño, resiliencia) asociados a la implementación de un sistema distribuido



## Contexto

Debido al cambio climático y en mayor medida, a las malas prácticas (intencionales y no intencionales) por parte de algunos ciudadanos, se han incrementado los incendios forestales en nuestro país. Los efectos son devastadores ya que acaban con la vida silvestre y la vegetación, los suelos pueden tardar siglos en recuperarse, se agotan las fuentes de agua, se aumenta la contaminación, etc. Por lo tanto, es importante monitorear de forma continua el estado de los suelos en zonas de bosque con el fin de detectar aumentos de

temperatura y cambios en las condiciones del ambiente para controlar a tiempo un posible incendio. Algunos de los factores que se deben monitorear son: temperatura del aire (oscila entre 11° y 29,4°), presencia de humo (Valor booleano) y humedad relativa (promedio 70%, una humedad inferior es indicativo de alguna anomalía).

En este proyecto se propone realizar un sistema distribuido utilizando los conceptos de Edge, Fog y Cloud computing<sup>1</sup>, en el que se simula la tolerancia a fallas y la captura y análisis de los tres parámetros previamente mencionados, de una zona boscosa de los cerros orientales de Bogotá. La arquitectura sugerida es la siguiente:

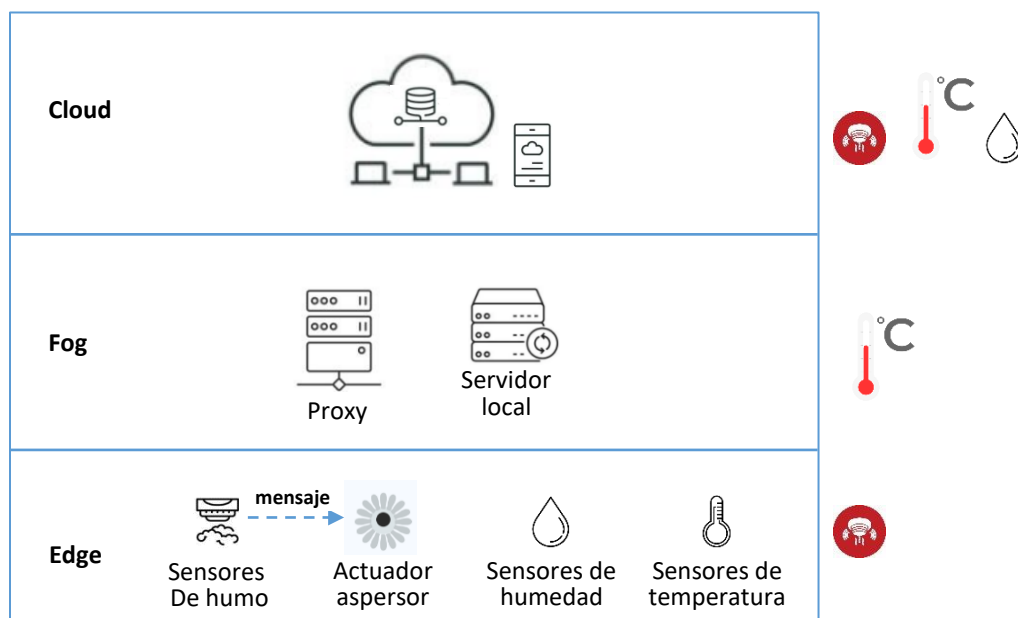


Fig. 1. Arquitectura general del sistema de detección de incendios

## Descripción del Sistema a Desarrollar

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

### Capa Edge Computing

- **Sensores de humo:** Se debe implementar un sensor de humo y levantar 10 instancias de sensores de humo, cada uno tomando muestras cada 3 segundos, las cuales se deben enviar a la capa de Fog Computing. El sensor de humo es un componente reactivo dentro de la arquitectura, es decir, tan pronto detecta una señal de humo el sensor realizará las siguientes acciones: a) Envía un mensaje directo a un sensor actuador que activará un aspersor. b) Envía el mensaje de alerta

<sup>1</sup> Cloud, Fog, and Edge Computing: A Software Engineering Perspective. doi: 10.1109/COMAPP.2018.8460443. url:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8460443&isnumber=8460181>

a la nube, a través del proxy de la capa *Fog Computing* (en el cloud se almacena toda la información sobre las alertas en las distintas capas para fines estadísticos) c) Genera al Sistema de Calidad que se encuentra en esta capa.

- **Actuador aspersor:** Es suficiente con la implementación de un único actuador. Si recibe un mensaje de un sensor de humo, debe activarse inmediatamente (representado en un mensaje de texto).
- **Sensores de temperatura:** Se debe implementar un sensor de temperatura y activar 10 instancias de este sensor. Cada sensor estará tomando muestras cada 6 segundos; estas mediciones se envían al Proxy correspondiente en la siguiente capa (**Fog**).
- **Sensores de humedad:** Como antes, se debe implementar un sensor de humedad y activar 10 instancias del mismo, con una toma de muestras cada 5 segundos. Estos datos se envían al Proxy de la siguiente capa.

En la tabla 1 se observan valores válidos para cada variable a medir.

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

Parámetro	Valor mínimo	Valor máximo
<b>Humo (Variable booleana)</b>	Falso	Verdadero
<b>Temperatura</b>	11°	29,4°
<b>Humedad</b>	70%	100%

Los sensores se simulan con hilos y al activar cada sensor, éstos deben recibir los siguientes argumentos: **Tipo de sensor y un archivo de configuración para cada uno**. En el archivo de configuración 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). Los valores fuera del rango deben generar alertas. Por ejemplo, si el archivo de configuración para los sensores de temperatura 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, para el sensor de temperatura:

25 15 -1 35 29 20 27 10 17 40

En el caso del sensor de humo, solo se producirán los valores verdadero y falso y valores erróneos.

Cada sensor simulará las medidas produciendo números aleatorios que atiendan las probabilidades descritas en el archivo de configuración. Dichas medidas se enviarán al Proxy utilizando los sockets (PUSH-PULL) cada  $t$  unidades de tiempo, según se explicó para cada tipo de sensor. Cada medición debe ir acompañada del tiempo (u hora) en la cual se produjo.

### Capa Fog Computing

**Proxy:** se encarga de la comunicación hacia la nube, es decir, todas las mediciones directas, cálculos y alertas generados en las capas *Edge* o *Fog* se envían a la nube a través del componente proxy. Al recibir los datos de los sensores, el proxy valida que no contengan errores

En el Fog computing, la temperatura se calcula con un promedio simple de los 10 sensores; este promedio se debe publicar por pantalla junto con la fecha de los valores tomados. En caso de que la temperatura supere el valor máximo del rango establecido para tal indicador (ver tabla), se debe activar una alerta enviando o un mensaje al sistema de calidad de la capa Fog. También debe enviar un SMS, correo, etc. a la nube para fines de almacenamiento.

El cálculo **diario** de la humedad se hace de acuerdo con la siguiente fórmula:

**$HRd_{ijh}$**  Es la humedad relativa del aire del día  $h$ , registrada en el sensor  $i$ , durante el mes  $j$ . Este indicador se calcula cada 5 segundos promediando el valor de los 10 sensores de humedad

El proxy reenvía las medidas calculadas de la humedad a la capa cloud donde se calculará la humedad relativa mensual.

### Capa Cloud Computing

Todas las alarmas detectadas por los sensores deben ser comunicadas y almacenadas en el *cloud* con propósitos estadísticos. En el caso especial de las medidas de humedad, estas se reciben del proxy y cada 20 segundos se calcularán las muestras mensuales. La humedad relativa se calcula con la siguiente fórmula:

$$HRm_{ij} = \frac{\sum_{h=1}^n HRd_{ijh}}{n}$$

- **$HRm_{ij}$**  Es la humedad relativa media por mes ( $j$ ) del aire, registrada en el sensor ( $i$ )

- **$HRd_{ijh}$**  Es la humedad relativa del aire del día h, registrada en el sensor i, durante el mes. Valor que viene de la capa Fog.
- **n:** Es el número de días registrados en el mes

Cada 20 segundos se calcula el resultado de los 4 resultados diarios para obtener la suma mensual

Si el promedio calculado de la humedad mensual está por debajo del límite, se genera una alerta que va al sistema de calidad de esta capa.

### Sistema de Calidad (SC)

La única función de este proceso es esperar las posibles alarmas que se generan en cada una de las capas e imprimir las alertas por pantalla. Debe existir un SC en cada nivel (*Edge, fog, cloud*)

### Detalles de Implementación

**Comunicación:** la comunicación entre los sensores y proxies se debe hacer con el modelo Pipeline de la librería ZeroMQ (siguiendo el patrón muchos a uno de los workers al sink). La comunicación entre los componentes de cada capa y su sistema de calidad se realizará utilizando el patrón **request-reply**. El sistema de calidad debe responder al emisor del mensaje, únicamente después de haber publicado la alerta por el canal correspondiente. La comunicación entre el proxy y la capa cloud se hace siguiendo un patrón request-reply.

**Fallas:** debe implementar una arquitectura tolerante a fallas, considerando **una posible falla del componente proxy. Los estudiantes deberán contar con un respaldo que realice las mismas funciones, el cual estará en ubicado una máquina diferente (Dirección IP diferente)**. Debe existir también otro proceso encargado de chequear que el proxy está o no en funcionamiento (*health check*). Cuando se detecte que 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, procesos proxy principal y proxy de respaldo en computadores diferentes. Los componentes de la capa cloud en otro computador.

## Rendimiento del Sistema

### Variables para medir el rendimiento del sistema

1. Cantidad de mensajes enviados (o tamaño en bytes enviado) en todo el sistema durante su funcionamiento. Tanto el total de mensajes, como el total enviado de una capa a otra.
2. Tiempos de comunicación entre el proxy y el nivel cloud. En este caso los estudiantes deberán mostrar:
  - a. Tiempos promedio de comunicación entre las dos capas. Desviación estándar asociada al promedio.
3. Cantidad total de alertas generadas por cada indicador.

Para la primera entrega los equipos deberán definir el procedimiento y las herramientas que utilizarán para valorar las métricas de desempeño citadas.

### Elementos que afectan el rendimiento del sistema

En la segunda y última entrega del proyecto los equipos medirán los dos tipos de métricas citados anteriormente (a y b), en los siguientes 3 escenarios:

Escenario 1	Escenario 2	Escenario 3
Las probabilidades generadas por los sensores de humedad y temperatura se encuentran en los siguientes rangos:  0,5 valores correctos 0,4 valores fuera del rango 0,1 Errores	Las medidas de temperatura, desde los sensores, se envían cada 4 segundos.	Se mantienen las periodicidades del escenario 2 y probabilidades del escenario 1, pero se produce una falla en el proxy del nivel fog.

Cada grupo tomará medidas con los tres escenarios y colocará el valor de las variables en tablas. Adicionalmente, realizará gráficos que permitan observar cómo es el cambio de las métricas en los tres escenarios planteados. A la luz de los resultados obtenidos, cada grupo realizará un análisis y concluirá sobre la necesidad (o no) de realizar mejoras en el diseño del sistema para una detección de alarmas más eficiente y para un mejor uso de recursos (en este caso el ancho de banda de la red).

A continuación, sugerimos algunas herramientas de monitoreo y generación de carga:

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/>)

# Entregas y Rubrica de Evaluación

## 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:

- Modelos del sistema (Arquitectónico, interacción, fallos y seguridad)
- Diseño del sistema: Diagrama de despliegue, Diagrama de componentes, 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 obtener las métricas de desempeño: paquetes enviados y tiempos de publicación de las alarmas.
- Implementación inicial: deben tener implementada al menos la generación de mediciones desde los sensores y su recepción por parte de los procesos proxy. Esto debe estar funcionando al menos 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 muestre:
  - La arquitectura implementada, distribución de los componentes entre máquinas y patrones de comunicación.
  - Tecnologías utilizadas para implementar el sistema y realizar las mediciones.



- Cómo 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 (acompañados por gráficos y tablas), conclusiones.
- Adicionalmente se le puede pedir que traiga el día de la sustentación las diapositivas que usó en su video.

## Calificación I Entrega

Valor 15%, evaluado sobre 5pts

Indicador	Valoración en puntos /5pts	Excelente	Competente	Deficiente
Informe	0.5	0.5	0.25	<0.25
Diseño del Proyecto	1.5	[1.5,1]	(1, 0.25]	< 0.25
Protocolo de pruebas	0.75	[0.75,0.5]	(0.5, 0.25]	< 0.25
Modelos del sistema	0.5	0.5	0.25	<0.25
Obtención de las métricas de rendimiento	0.25	0.25	0.15	<0.15
Implementación Inicial	1.5	[1.5,1]	(1, 0.5]	< 0.5
total	5ptos			

Indicador	Excelente	Competente	Deficiente
Informe	La presentación del informe es impecable, sin problemas de ortografía o redacción	El informe tiene fallas menores en la presentación, ortografía y/o redacción	El informe tiene fallas importantes en la presentación, ortografía y/o redacción
Diseño del Proyecto	Se presentan todos los artefactos de diseño exigidos en el enunciado: diagrama de componentes,	Los diagramas incompletos o algunos diagramas están incorrectos.	No se realizan los diagramas exigidos en el enunciado o los diagramas presentados están incorrectos. No se considera TODO el

	<p>diagrama de clases, diagrama de secuencia y diagrama de despliegue. Se incluyen todos los componentes del sistema final, incluyendo tolerancia a fallas y persistencia. Los diagramas están correctos.</p>		<p>sistema que se va a implementar en el diseño.</p>
Protocolo de pruebas	<p>En el protocolo presentado por los estudiantes, las pruebas son suficientes para evaluar la funcionalidad de la aplicación con y sin la presencia de fallas.</p>	<p>En el protocolo presentado no se contemplan todas las pruebas importantes. Está incompleto.</p>	<p>Los estudiantes no presentan protocolo de pruebas funcionales.</p>
Modelos de sistema	<p>Se hace una descripción de los modelos fundamentales adaptados al proyecto.</p>	<p>Se describen parcialmente los modelos fundamentales adaptados al problema a resolver.</p>	<p>No se presenta ni se describen los modelos fundamentales o la descripción no está relacionada con el proyecto a resolver.</p>
Obtención de las métricas de rendimiento	<p>Se describe de forma clara y completa las herramientas y metodología para obtener el valor de las métricas.</p>	<p>En el informe falta alguno de los siguientes puntos o están mencionados parcialmente:</p> <ul style="list-style-type: none"> <li>- Herramientas de medición que van a usar</li> <li>- Metodología para obtener el valor de las métricas.</li> </ul>	<p>No se mencionan en el informe ni las herramientas ni la metodología para obtener el valor de las métricas de rendimiento.</p>
Implementación Inicial	<p>Se implementa la generación de los tres tipos de mediciones desde los sensores. Se puede evidenciar que cada proxy</p>	<p>Se implementa la generación de las medidas desde los sensores, pero de forma parcial, es decir, se observa alguna de las siguientes deficiencias (máximo 2 de 4):</p>	<p>La implementación inicial es bien deficiente, se observan falencias en 3 ó 4 de los puntos</p>

	recibe la información a la que está suscrito. Los procesos se invocan de la forma como fue especificado.	a) El sistema solo funciona en un computador b) Los procesos no se levantan con los parámetros solicitados. c) No está correctamente implementado el patrón publicador/suscriptor. d) No se implementa correctamente la generación de medidas desde ellos sensores (atendiendo las probabilidades sugeridas)	mencionados en la celda anterior (a-d)
--	--	---	--

## Calificación II Entrega

**Valor 25%**

**Informe de rendimiento 10% (Evaluado sobre 5ptos)**

**Resto de la entrega (corrida, sustentación, etc) 15%. Evaluado sobre 5ptos**

## Rúbricas

### Informe de Rendimiento

	Valoración en puntos	Excelente	Competente	Deficiente
Informe: presentación. Ortografía y redacción	1	[1,0.75]	(0.75, 0.5]	< 0.5
Presentación de los datos: datos presentados en tablas y gráficos. Se grafican las variables solicitadas, los gráficos están contruidos correctamente.	2	[2,1]	(1, 0.5]	< 0.5
Análisis de los resultados: el	2	[2,1]	(1, 0.5]	< 0.5

grupo comenta los resultados de interés y deriva conclusiones cónsonas con estos resultados				
	5			

### Descripción

	Valoración en puntos	Excelente	Competente	Deficiente
Informe: presentación. Ortografía y redacción	1	La presentación del informe es impecable sin problemas de ortografía o redacción	El informe tiene fallas menores en la presentación, ortografía y/o redacción	El informe tiene fallas importantes en la presentación, ortografía y/o redacción
Presentación de los datos: datos presentados en tablas y gráficos.	2	Se grafican todas las variables solicitadas, los gráficos están contruidos correctamente (títulos, unidades, etc)	Faltan algunos gráficos o, están completos, pero faltan elementos que contribuyen a su comprensión: qué se está graficando, cuáles son las unidades de medida, etc.	No se realizan gráficos o se realizan muy pocos gráficos. La presentación de los gráficos es bastante deficiente.
Análisis de los resultados: el grupo comenta los resultados de interés y deriva conclusiones cónsonas con estos resultados	2	Los estudiantes comentan los resultados obtenidos y presentan conclusiones sobre la arquitectura del sistema y los patrones de comunicación.	Los estudiantes comentan los resultados obtenidos pero no presentan conclusiones sobre la arquitectura del sistema y los patrones de comunicación.	Los estudiantes no comentan los resultados obtenidos o los comentarios son muy deficientes.

### Funcionamiento del Sistema, Sustentación

Indicador	Valoración en puntos /30	Excelente	Competente	Deficiente
<b>Capa Edge:</b> Implementación de los sensores, actuador, sistema de calidad	0.5	0.5	0.25	< 0.25
<b>Capa Fodge:</b> funcionamiento del proxy y sistema de calidad	0.75	[0.75, 0.5]	(0.5, 0.25]	< 0.25
Capa cloud: alertas de humedad, sistema de calidad, persistencia	0.75	[0.75, 0.5]	(0.5, 0.25]	< 0.25
Corrida en 3 máquinas	0.75	[0.75, 0.5]	(0.5, 0.25]	< 0.25
Tratamiento de Fallas del Proxy	0.5	0.5	0.25	< 0.25
Código	0.25	0.25	(0.25, 0.15]	< 0.15
Protocolo de pruebas funcionales	0.5	0.5	0.25	< 0.25
Sustentación, video	1	[1,0.75]	(0.75, 0.5]	< 0.5
Total	5pts			

#### Descripción

<b>Capa Edge:</b> Implementación de los Sensores, actuador, Sistema de calidad	Los sensores y el actuador están implementados correctamente y según las especificaciones del enunciado. Se puede evidenciar la corrida de los 10 sensores de cada tipo. Se disparan las alarmas correctamente.	Los sensores están implementados correctamente y, siguen la mayoría de las especificaciones del enunciado. En la ejecución se evidenció alguna de las siguientes fallas (máximo 2): <ul style="list-style-type: none"> <li>- No corrieron los 10 sensores de cada tipo.</li> <li>- No funcionó el actuador correctamente.</li> </ul>	Los sensores están implementados, pero hay diferencias importantes entre las especificaciones del enunciado y lo que implementaron los estudiantes. La información no llega a los proxies de la forma correcta.
---	---	--	---

		<ul style="list-style-type: none"> <li>- No se disparan las alarmas correctamente.</li> <li>- Las medidas no se producen con la probabilidad especificada.</li> <li>- Los sensores no se inician según las especificaciones del enunciado.</li> </ul>	
<b>Capa Fog:</b> Proxy Sistema de Calidad	El proxy recibe la información adecuada de los sensores, almacena los datos y genera las alarmas correspondientes. Se envía de forma correcta la información a la siguiente capa. El proxy se implementa según las especificaciones del enunciado.	El proxxt recibe la información correctamente de los sensores. No obstante, alguno de los siguientes aspectos está parcial, incorrecto o incompleto: <ul style="list-style-type: none"> <li>- Generación de alarmas</li> <li>- Argumentos de inicialización de este tipo de procesos.</li> <li>- Envío de información a la siguiente capa.</li> </ul>	La implementación del Proxy es deficiente en más de uno de los siguientes aspectos: <ul style="list-style-type: none"> <li>- Recepción de las medidas</li> <li>- Generación de alarmas</li> <li>- Argumentos de inicialización de este tipo de procesos.</li> <li>- Envío de información a la siguiente capa.</li> </ul>
Capa cloud	El sistema recibe las medidas de las capas anteriores, detecta las alarmas, realiza los cálculos correspondientes y se implementa la persistencia.	La capa se implementa, pero 1 ó 2 de las siguientes funcionalidades no está del todo correcta: <ul style="list-style-type: none"> <li>- Tratamiento de alarmas</li> <li>- Cálculos mensuales de los diferentes indicadores</li> <li>- Persistencia</li> <li>- Manejo de alarmas.</li> </ul>	La capa se implementa, pero hay deficiencias en tres o cuatro de las siguientes funcionalidades: <ul style="list-style-type: none"> <li>- Tratamiento de alarmas</li> <li>- Cálculos mensuales de los diferentes indicadores</li> <li>- Persistencia</li> <li>- Manejo de Alarmas</li> </ul>

Corrida en 3 máquinas	El sistema funciona en al menos tres computadoras (o máquinas virtuales) según la arquitectura sugerida en el enunciado del proyecto	El sistema funciona solo en dos computadoras y/o máquinas virtuales	Todos los componentes del sistema se instalan y funcionan en una sola computadora.
Tratamiento de Fallas del proxy	Ante la falla del proxy la inicialización de un proceso homólogo funciona de forma correcta. La falla se detecta de forma automática usando un patrón, como por ejemplo, el health check. Los estudiantes explican el o los patrones de resiliencia implementados	Ante la falla del proxy se levanta un proceso homólogo que asume de forma correcta sus funciones. La falla no se detecta de forma automática. Los estudiantes no explican claramente los patrones de resiliencia implementados.	No se implementa la tolerancia a fallas de este componente.
Código	El código está bien estructurado, bien indentado y documentado	El código es funcional pero no está debidamente documentado	El código es deficiente, sin orden ni documentación
Protocolo de pruebas funcionales	Se hacen todas las pruebas necesarias para evaluar la funcionalidad del sistema	Faltaron algunas pruebas funcionales	Las pruebas funcionales son insuficientes.
Sustentación, video	Se siguen todas las reglas dadas para la sustentación, se presentan todos los elementos importantes del sistema y se responden de forma adecuada todas las inquietudes del evaluador. Se entrega el video en el que se evidencia claramente la arquitectura de la aplicación, su funcionamiento y los	Hay alguna <i>falla menor</i> en uno o dos de los siguientes elementos: <ul style="list-style-type: none"> <li>- Seguimiento de las reglas dadas para la sustentación</li> <li>- Presentación de las funcionalidades del sistema</li> <li>- Respuesta correcta a todas las preguntas del evaluador.</li> <li>- El video no ilustra de</li> </ul>	Hay <i>fallas</i> importantes en 3 o más <i>de los</i> siguientes elementos: <ul style="list-style-type: none"> <li>- Seguimiento de las reglas dadas para la sustentación</li> <li>- Presentación de las funcionalidades del sistema</li> <li>- Respuesta correcta a todas las preguntas del evaluador.</li> </ul>

	elementos de tolerancia a fallas.	manera suficiente la arquitectura, el funcionamiento de la aplicación o la tolerancia de las fallas.	- El video es deficiente o no se adjunta.
--	-----------------------------------	--	---