

---

# ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (SRS)

---

MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA  
NODOS EQUIVALENTES



**AUTOR:**

Ing. Juan Carlos Zárate Trejo – Programador/Diseñador

**STAKEHOLDERS:**

Dr. Félix Francisco Ramos Corchado



28 DE ABRIL DE 2023

CINVESTAV- Unidad GDL

SDD  
MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS  
EQUIVALENTES

## Contenido

<b>1.</b>	<b><i>Introducción</i></b> .....	<b>2</b>
1.1	Alcance .....	2
1.2	Definiciones, acrónimos y abreviaturas .....	2
1.3	Referencias: .....	2
1.4	Visión general del documento .....	3
<b>2.</b>	<b><i>Descripción General</i></b> .....	<b>3</b>
2.1	Perspectiva del producto: .....	3
2.2	Funciones del producto: .....	3
2.3	Características de los usuarios: .....	4
2.4	Restricciones generales: .....	4
2.5	Suposiciones y dependencias:.....	4
<b>3.</b>	<b><i>Requerimientos</i></b> .....	<b>4</b>
3.1	Requerimientos funcionales .....	4
3.2	Requisitos No Funcionales: .....	5
3.7	Restricciones de diseño: .....	6
3.8	Atributos de calidad del software: .....	6
<b>4.</b>	<b><i>Resumen de requerimientos implementados</i></b> .....	<b>7</b>

## 1. Introducción

El propósito de este documento es definir los requerimientos del sistema de comunicación entre nodos, el cual permitirá la conexión y comunicación de nodos en una red distribuida.

### 1.1 Alcance

El sistema de comunicación entre nodos se enfoca en la conexión y comunicación entre nodos en una red distribuida. El sistema permitirá la auto asignación de roles entre los nodos miembros, verificación de la presencia del nodo maestro en la red, interacción por medio de claves de cifrado asimétrico, recuperación de estados de los nodos después de eventos críticos, y la ejecución en tiempo real.

### 1.2 Definiciones, acrónimos y abreviaturas

- **Nodo:** dispositivo o entidad que se conecta y comunica en una red distribuida.
- **Autoasignación:** proceso por el cual los nodos miembros pueden definir sus propios roles en la red.
- **Nodo maestro:** nodo que actúa como punto de control y coordinación en la red.
- **Claves de cifrado asimétrico:** sistema de cifrado en el que se utilizan dos claves diferentes para cifrar y descifrar información.
- **Eventos críticos:** situaciones en las que se produce una desconexión o pérdida parcial de comunicación en la red.
- **API:** interfaz de programación de aplicaciones.

### 1.3 Referencias:

- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

### **1.4 Visión general del documento**

El presente documento se divide en dos secciones principales: los requerimientos funcionales y los requerimientos no funcionales. En la sección de requerimientos funcionales, se describen los comportamientos y funcionalidades que el sistema debe ser capaz de ejecutar. Por otro lado, en la sección de requerimientos no funcionales, se describen los aspectos no relacionados con las funcionalidades que el sistema debe cumplir, como el rendimiento y la escalabilidad.

## **2. Descripción General**

### **2.1 Perspectiva del producto:**

El sistema se desarrollará como un conjunto de nodos miembros, que funcionarán como un núcleo de comportamiento de bajo nivel para aplicaciones más grandes a través de una API o interfaz de programación. Los nodos miembros interactuarán entre sí de manera distribuida para llevar a cabo tareas complejas y coordinar recursos. El sistema será capaz de ejecutarse en tiempo real, considerando los procesos temporizados.

### **2.2 Funciones del producto:**

El sistema permitirá la auto asignación de roles entre los nodos miembros y será capaz de verificar si los miembros están en línea. La interacción entre módulos debe ser débilmente acoplada y las tareas entre módulos deben ser mutuamente excluyentes. Los nodos deberán ser capaces de verificar la presencia del nodo maestro en la red y el nodo maestro debe ser capaz de comprobar si el nodo con el que se está comunicando es legítimo en base al token con el que dicho nodo debió darse de alta en la red. Además, el sistema deberá ser capaz de recuperar los estados de los nodos después de eventos críticos y permitir la restauración o expulsión de nodos para reducir el uso de recursos de red.

### **2.3 Características de los usuarios:**

El sistema podrá ser usado por cualquier usuario con el conocimiento y las herramientas de desarrollo adecuadas, como lenguajes de programación orientados a objetos, fundamentos de tecnologías de comunicación por red como sockets, RPC y RMI.

### **2.4 Restricciones generales:**

El sistema se ejecutará en equipos con especificaciones discretas, incluyendo 1 GB de RAM y 1 CPU a 1.4 GHz. Además, el sistema estará enfocado en aplicaciones de escritorio y dispositivos terminales basados en Linux.

### **2.5 Suposiciones y dependencias:**

El sistema en cuestión no requiere medidas de seguridad específicas para los usuarios, a excepción de las especificaciones de cifrado de mensajes entre nodos. La escalabilidad del sistema se medirá en función de la capacidad de la red, lo que garantizará un alto rendimiento y un desempeño óptimo en todo momento. El sistema deberá estar disponible constantemente, a menos que ocurran eventos críticos, en cuyo caso deberá emitir alertas de estado para notificar a los usuarios de las situaciones que puedan afectar la disponibilidad. En cuanto a la facilidad de uso, esta dependerá principalmente del conocimiento de los detalles del manual de usuario y la experiencia que los usuarios tengan con los temas relacionados. No se requieren especificaciones legales para el sistema, aunque se espera que implemente rutinas bajo una perspectiva algorítmica eficiente, tanto en términos de tiempo de ejecución como de memoria y energía.

## **3. Requerimientos**

### **3.1 Requerimientos funcionales**

- RF-001: El sistema debe permitir la auto asignación de roles entre los nodos miembros, verificar la disponibilidad en línea de los miembros, tolerar conexiones inestables y mantener la seguridad del flujo de información.

## MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS EQUIVALENTES

- RF-002: El sistema debe ser accesible a cualquier usuario con conocimiento y herramientas de desarrollo apropiadas.
- RF-003: El sistema debe contar con una interfaz de alto nivel para ingresar la información de los parámetros de inicio en formato estándar.
- RF-004: La interacción entre módulos debe ser débilmente acoplada y las tareas entre módulos deben ser mutuamente excluyentes.
- RF-005: Los nodos deben verificar la presencia del nodo maestro en la red y el nodo maestro debe comprobar la legitimidad del nodo en función del token utilizado para darse de alta en la red.
- RF-006: Los errores en los nodos deben permitir su restauración o expulsión de la red para reducir el uso de los recursos de red y deben manifestarse como registros en la interfaz.
- RF-007: El sistema debe interactuar mediante claves de cifrado asimétrico basado en el perfil de roles establecido en el modelo.
- RF-008: El sistema debe recuperar los estados de los nodos después de eventos críticos, entendidos como cualquier detonante de desconexión o pérdida parcial de comunicación.
- RF-009: El sistema debe ejecutarse en tiempo real, considerando los procesos temporizados.
- RF-010: El sistema debe funcionar como un núcleo de comportamiento de bajo nivel sobre aplicaciones más grandes a través de una API o interfaz de programación.

**3.2 Requisitos No Funcionales:**

- RNF-001: El sistema debe ejecutarse en equipos con especificaciones mínimas de 1GB RAM y 1 CPU @ 1.4GHz.
- RNF-002: El sistema debe implementar cifrado de mensajes entre nodos para garantizar la seguridad de la información.
- RNF-003: El grado de escalabilidad del sistema debe estar dado en función de la capacidad de la red.

## MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS EQUIVALENTES

- RNF-004: El sistema debe estar disponible en todo momento, salvo en eventos críticos, para los cuales debe proporcionar alertas de estado.
- RNF-005: La facilidad de uso del sistema estará dada únicamente por el conocimiento del manual de usuario y la experiencia del usuario con los tópicos correspondientes.
- RNF-006: El sistema está diseñado para aplicaciones de escritorio y dispositivos terminales basados en Linux.
- RNF-007: El sistema debe tener una descripción y gestión claras.
- RNF-008: No existen requerimientos legales específicos para el sistema.
- RNF-009: El sistema debe implementar rutinas algorítmicas eficientes en términos de tiempo de ejecución, memoria y energía.
- RNF-010: No hay requisitos mínimos para el sistema.

**3.7 Restricciones de diseño:**

- El sistema debe ser diseñado para funcionar en una red local de tamaño mediano, con un número máximo de nodos estimado de 50.
- El sistema debe ser fácilmente configurable por parte del administrador de red para permitir la inclusión y exclusión de nodos de la red.
- La interfaz de usuario debe ser diseñada de manera intuitiva y amigable para el usuario, para facilitar su uso y reducir la curva de aprendizaje.
- El sistema debe estar diseñado para ser modular y escalable, permitiendo la adición de nuevas funcionalidades sin afectar la estructura existente.
- El sistema debe ser diseñado con una arquitectura distribuida, para permitir su operación en diferentes nodos de la red, evitando así la sobrecarga en un solo nodo.

**3.8 Atributos de calidad del software:**

- **Fiabilidad:** El sistema debe ser capaz de funcionar de manera confiable en un entorno de red, tolerando la pérdida de paquetes y la desconexión de nodos sin afectar la operación de la red.

## MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS EQUIVALENTES

- **Escalabilidad:** El sistema debe ser diseñado para ser escalable en función del tamaño de la red y del número de nodos, permitiendo la adición y eliminación de nodos de manera transparente al usuario.
- **Seguridad:** El sistema debe ser seguro, garantizando la privacidad y la integridad de la información transmitida entre los nodos de la red.
- **Rendimiento:** El sistema debe ser capaz de procesar grandes volúmenes de datos en tiempo real, sin afectar su desempeño ni la calidad del servicio.
- **Mantenibilidad:** El sistema debe ser fácilmente mantenible y modificable por parte de los desarrolladores, permitiendo la corrección de errores y la inclusión de nuevas funcionalidades sin afectar el funcionamiento del sistema.

### 4. Resumen de requerimientos implementados

ID	Descripción	Estado	Observaciones
<b>RF-001</b>	El sistema debe permitir la auto asignación de roles entre los nodos miembros, verificar la disponibilidad en línea de los miembros, tolerar conexiones inestables y mantener la seguridad del flujo de información.	Alcanzado	
<b>RF-002</b>	El sistema debe ser accesible a cualquier usuario con conocimiento y herramientas de desarrollo apropiadas sobre tecnologías de red y lenguajes POO.	Alcanzado	
<b>RF-003</b>	El sistema debe contar con una interfaz de alto nivel	No Alcanzado	No se realizó la implementación de alguna



MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS EQUIVALENTES

	para ingresar la información de los parámetros de inicio en formato estándar.		interfaz que permita el ingreso de parámetros al sistema. La configuración se realiza a nivel de script.
<b>RF-004</b>	La interacción entre módulos debe ser débilmente acoplada y las tareas entre módulos deben ser mutuamente excluyentes.	Alcanzado	
<b>RF-005</b>	Los nodos deben verificar la presencia del nodo maestro en la red y el nodo maestro debe comprobar la legitimidad del nodo en función de algún token legítimo.	Alcanzado	
<b>RF-006</b>	Los errores en los nodos deben permitir su restauración o expulsión de la red para reducir el uso de los recursos de red y deben manifestarse como registros en la interfaz.	Alcanzado	
<b>RF-007</b>	El sistema debe interactuar mediante claves de cifrado asimétrico basado en el perfil de roles establecido en el modelo.	No Alcanzado	El sistema no presenta ningún método de cifrado de mensajes.
<b>RF-008</b>	El sistema debe recuperar los estados de los nodos después de eventos críticos, entendidos como cualquier detonante de desconexión o	Parcialmente Alcanzado	Los estados de los nodos maestro, de momento, se consideran irrecuperables dado que los nodos regulares

MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS EQUIVALENTES

	pérdida parcial de comunicación.		no almacenan datos de los primeros.
<b>RF-009</b>	El sistema debe ejecutarse en tiempo real, considerando los procesos temporizados.	Parcialmente Alcanzado	Dado la temporización de procesos y la necesidad de sincronía entre ellos en ciertos eventos, los tiempos de respuesta pueden variar en momentos puntuales.
<b>RF-010</b>	El sistema debe funcionar como un núcleo de comportamiento de bajo nivel sobre aplicaciones más grandes a través de una API o interfaz de programación.	Parcialmente Alcanzado	Si bien, se puede construir una aplicación que implemente el sistema como base, no existe una API o interfaz de comunicación predeterminada.
<b>RNF-001</b>	El sistema debe ejecutarse en equipos con especificaciones mínimas de 1GB RAM y 1 CPU @ 1.4GHz.	Alcanzado	
<b>RNF-003</b>	El grado de escalabilidad del sistema debe estar dado en función de la capacidad de la red.	Alcanzado	
<b>RNF-004</b>	El sistema debe estar disponible en todo momento, salvo en eventos críticos, para los cuales debe proporcionar alertas de estado.	Alcanzado	
<b>RNF-005</b>	El sistema está diseñado para aplicaciones de escritorio y dispositivos	Alcanzado	

MODELO DE AUTOASIGNACIÓN DE ROLES EN SISTEMAS DISTRIBUIDOS PARA NODOS EQUIVALENTES

	terminales basados en Linux.		
<b>RNF-006</b>	El sistema debe tener una descripción y gestión claras, y contar con un manual de usuario.	Alcanzado	
<b>RNF-007</b>	El sistema debe implementar rutinas algorítmicas eficientes en términos de tiempo de ejecución, memoria y/o energía.	Parcialmente Alcanzado	El algoritmo de consenso implementa un procedimiento de complejidad al menos cuadrática, por lo que puede llegar a inundaciones de paquetes en la red.