

MANUAL DE INSTALACIÓN DEL SISTEMA DE MONITOREO MEDICO



Universidad
del Cauca

**Juan David García
Andrés Camilo Meneses**

Ing. Daniel Paz

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Programa de Ingeniería de Sistemas
Curso: Sistemas Distribuidos
Popayán, Noviembre de 2019**

TABLA DE CONTENIDO

1. Introducción
2. Requerimientos del Sistema: Sistema Operativo, Tecnología utilizada, Lenguaje de programación
3. Descripción de la documentación entregada
4. Descripción de la estructura de directorios, descripción de los archivos de soporte
5. Descripción de los archivos fuente (Interfaces, archivos cliente, archivos servidor)
6. Descripción del proceso de compilación y ejecución
7. Descripción del Manejo del Instalador.

1. Introducción

Este manual permitirá la compilación, instalación y ejecución del sistema de monitoreo medico, guiando al usuario (técnico de instalación) para una correcta instalación en sistemas operativos Linux.

2. Requerimiento del sistema

- **Sistema Operativo:** El sistema ha sido probado en sistemas operativos GNU/Linux con kernel 4.0 o superior (también para kernel genérico).
- **Tecnología utilizada.** El sistema utiliza tecnología RPC (Remote Procedure Call), escrito en el editor de texto Visual Code, compilador GCC, la herramienta de gestión de dependencias make, rpcbin para el uso de RPC en linux.
- **Lenguaje de programación.** Lenguaje C.

3. Descripción de la documentación entregada. Este manual contiene la información necesaria para una correcta compilación y ejecución del programa en sistemas Linux. El manual va dirigido a un usuario que tiene conocimientos básicos de sistemas Linux y lenguaje C. Ademas se anexan el manual técnico el cual contiene la definición de los requisitos, diagramas de secuencia y la descripción de las estructuras utilizadas y el manual de usuario el cual describe lo que puede hacer el usuario.

4. Descripción de la estructura de directorios, descripción de los archivos de soporte. Los directorios están organizados de la siguiente manera.

- **Directorio General:** Contiene las carpetas referentes al host cliente, servidor de alertas y servidor de notificaciones ademas de el archivo (script) autogen.sh el cual permite compilar y ejecutar un cliente, el servidor de alertas y el servidor de notificaciones automáticamente, cada uno en un terminal distinto.

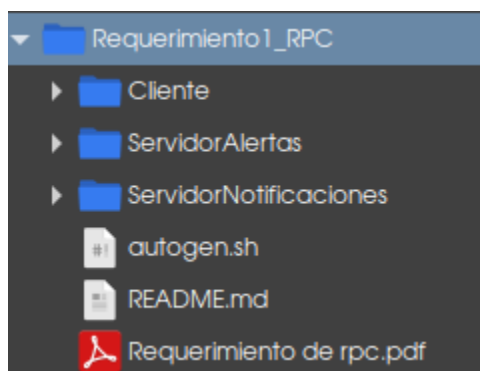


Figura 1: Estructura general de los directorios

- **Directorio Cliente:** Contiene el código fuente necesario para el registro de un paciente y la lectura de los sensores que se encuentren conectados a la aplicación permitiendo el envío del resultado al servidor de alertas haciendo uso de la tecnología RPC. Además, contiene el archivo Makefile para la compilación de todos los archivos fuente.

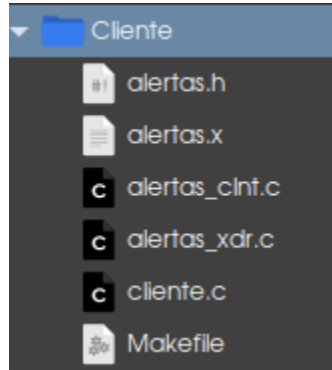


Figura 2: Estructura del directorio cliente

- **Directorio ServidorAlertas:** Contiene el código fuente necesario para la gestión de alertas con la información de los indicadores proporcionada por el host cliente, en caso de que algún indicador se encuentra fuera de los rangos normales (dependiendo de la edad del paciente) se generará una alerta que es almacenada en el archivo historialDeAlertas.txt y se realiza el envío de la alerta al servidor de notificaciones para ser mostrada por el mecanismo previsto para el servidor de notificaciones. Además, contiene el archivo Makefile para la compilación de todos los archivos fuente.

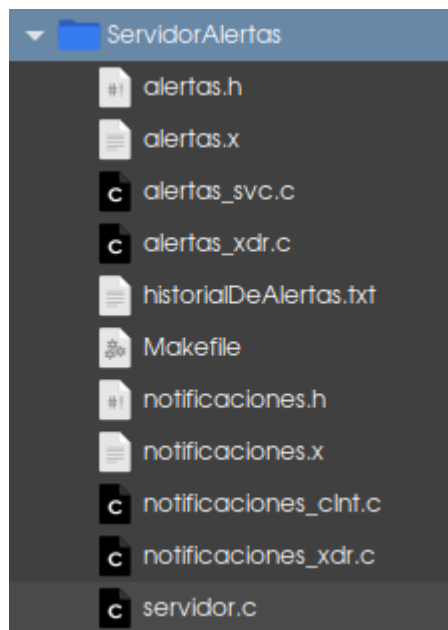


Figura 3: Estructura del directorio ServidorAlertas

El **Archivo HistorialDeAlertas.txt** almacenara el numero de habitación, nombres y apellidos, fecha y hora de la alerta y la puntuación de las alertas generadas.

- **Directorio ServidorNotificaciones:** Este directorio contiene todos los archivos fuente para la gestión de notificaciones sobre las alertas enviadas por el servidor de alertas. Además, contiene el archivo Makefile para la compilación de todos los archivos fuente.

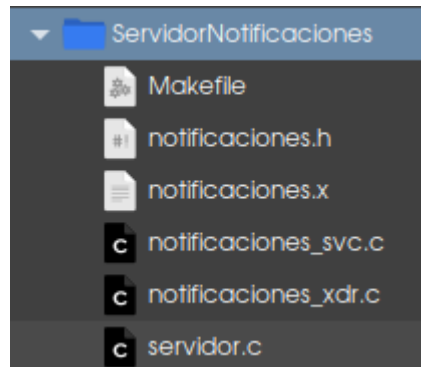


Figura 4: Estructura del directorio ServidorNotificaciones

5 . Descripción de los archivos fuente:

- **Interfaces:** alertas.x. En esta interface se define el método **enviarIndicadores** que va a ser invocado desde el cliente hacia **servidorAlertas** además se definen las estructuras **Paciente** que encapsula la información personal de un paciente como su nombre, apellidos, edad, numero de habitación y el resultado de las lecturas de los sensores, **Indicadores** encapsula la información leída por los sensores. Notificaciones.x. En esta interface se define el procedimiento **enviarNotificacion** que va a ser invocado desde el servidor de alertas hacia el servidor de notificaciones, además se definen las estructuras **infoPaciente** encargada de encapsular la información de un paciente como lo son sus nombres, apellidos, edad, numero de habitación y la fecha y hora de la notificación, **IndicadoresAlerta** utilizada para la gestión de los indicadores que generaron la alerta contiene el nombre del indicador y el valor, **UltimasAlertas** encargada de encapsular la información de las ultimas cinco alertas generadas a un paciente con la información de la fecha y hora de la alerta y la puntuación.
- **Archivos Cliente:**

- *alertas_clnt.c*: Contiene el código fuente necesario para que el cliente de alertas se comuniquen con el servidor de alertas.
- *alertas.h*: Header donde se declaran los procedimientos remotos y los datos que manipulará el cliente de alertas y el servidor de alertas.
- *alertas_xdr.c*: Este archivo contiene la nacionalización (reservado de memoria) de los datos que fueron declarados en la interfaz alertas.x.
- *cliente.c*: Archivo en el cual se encuentra la lógica principal del programa, permitiendo el registro de un paciente como la lectura y envío de indicadores generados por los sensores.
- *notificaciones_clnt.c*: Contiene el código necesario para la conexión entre el servidor de alertas y el servidor de notificaciones.
- *notificaciones_xdr.c*: Este archivo se encarga de la reserva de memoria para los datos declarados en la interfaz notificaciones.x.
- *notificaciones.h*: Header donde se declaran en lenguaje C los datos y procedimientos declarados en la interfaz notificaciones.h en formato xdr.
- **Archivos servidor:**
 - *alertas_svc.c*: Contiene la lógica necesaria para permitir que un cliente de alertas se conecte para permitir la compilación.
 - *Notificaciones_svs.c*: Contiene el código que permite la comunicación entre el servidor de alertas y el servidor de notificaciones.
 - *Alertas_xdr.x*, *alertas.h*, *notificaciones.h* y *notificaciones_xdr.c* cumplen la misma función descrita anteriormente.

6. Descripción del proceso de compilación y ejecución. Para compilar el programa en conjunto utilice el archivo autogen.sh siguiendo los siguientes pasos:

- ✓ Abra un terminal en la carpeta del proyecto.
- ✓ Ejecute el siguiente comando como administrador *chmod +x autogen.sh* para que otorgar permisos de ejecución al archivo autogen.sh.

- ✓ Ejecute el siguiente comando para iniciar el proceso de compilación y ejecución
`./autogen.sh`

Nota: El archivo autogen.sh solo sera ejecutado correctamente si se tiene instalado el gestor de dependencias make y Gnome-Terminal de lo contrario compile haciendo make dentro de cada directorio y ejecute cada paquete por separado.

7. Descripción del archivo de instalación. El programa no cuenta con archivo de instalación.