Contenido

[1 Introducción y objetivos 3](#_Toc94694165)

[2 Diseño Hardware 3](#_Toc94694166)

[2.1 Esquema electrónico del dispositivo 3](#_Toc94694167)

[2.2 Esquema de redes 3](#_Toc94694168)

[3 Diseño Software 4](#_Toc94694169)

[3.1 Flujo del programa 4](#_Toc94694170)

[3.1.1 Bot de Telegram 4](#_Toc94694171)

[3.2 Descripción del funcionamiento 7](#_Toc94694172)

[3.2.1 Bot de Telegram 7](#_Toc94694173)

[3.3 Robustez 9](#_Toc94694174)

[3.4 Posibles mejoras y trabajo futuro 9](#_Toc94694175)

[3.5 Librerías 9](#_Toc94694176)

[4 Resultados y conclusiones 9](#_Toc94694177)

[5 Manual de usuario 10](#_Toc94694178)

[5.1 Manual del administrador de SkapaLab 10](#_Toc94694179)

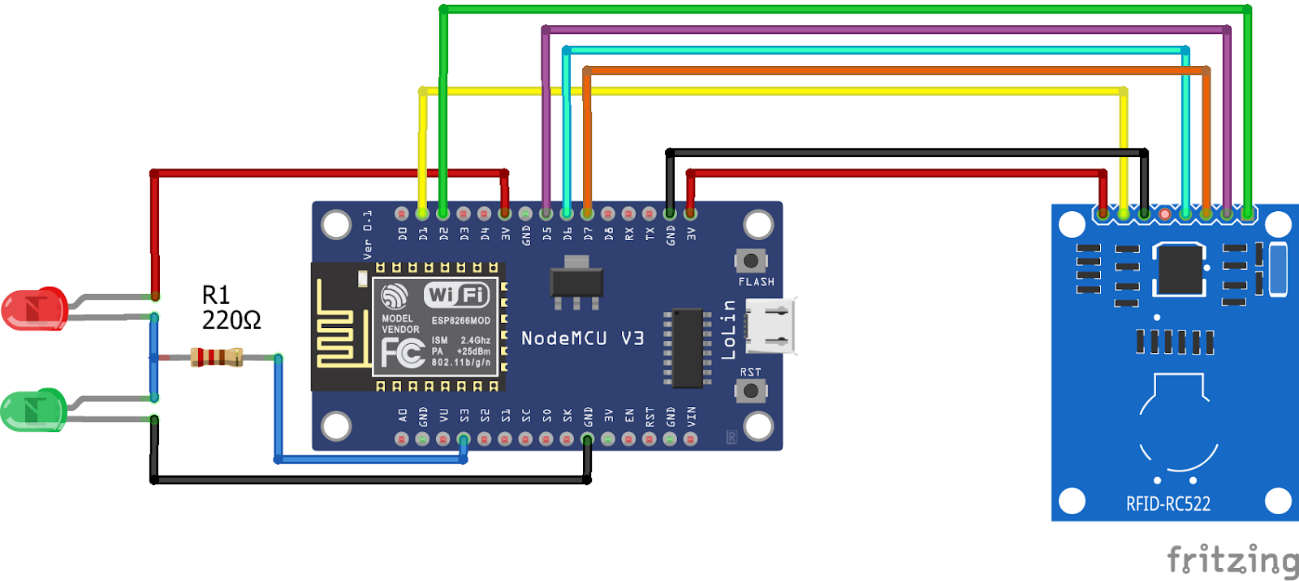
[5.2 Manual del usuario miembro de SkapaLab 10](#_Toc94694180)

[6 Ficheros entregados 11](#_Toc94694181)

# Introducción y objetivos

# Diseño Hardware

## Esquema electrónico del dispositivo



## Esquema de redes

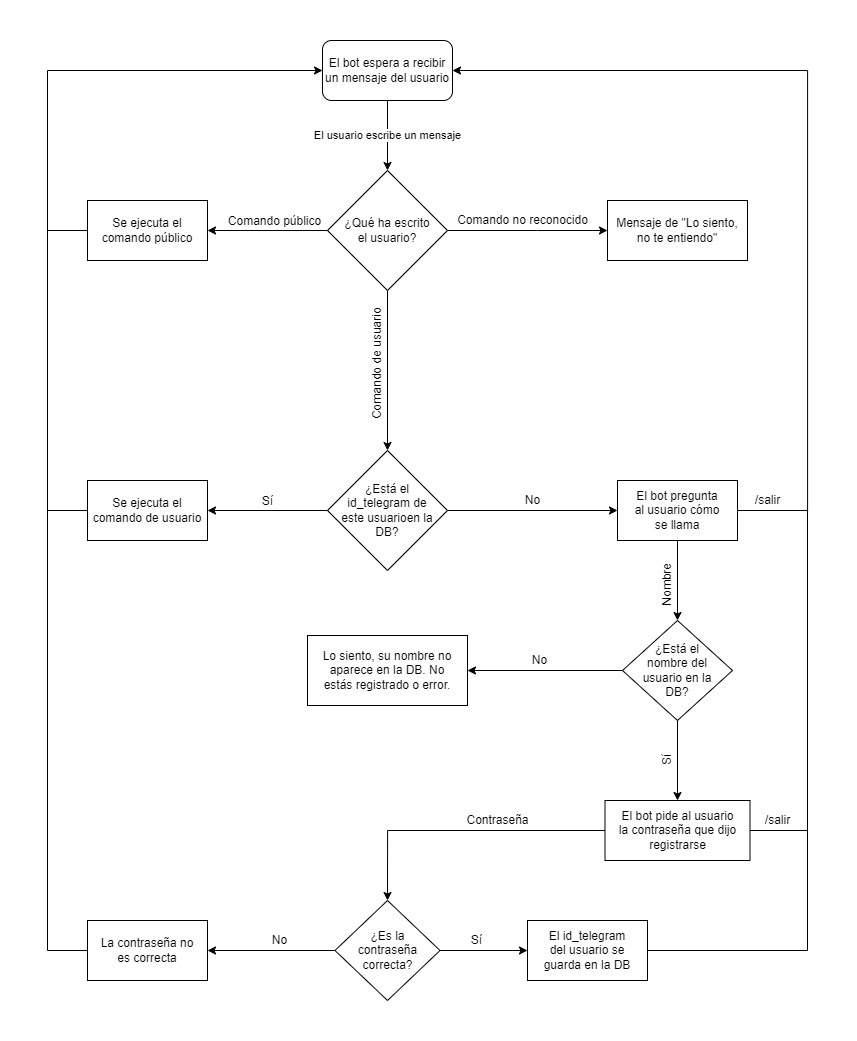
# Diseño Software

## Flujo del programa

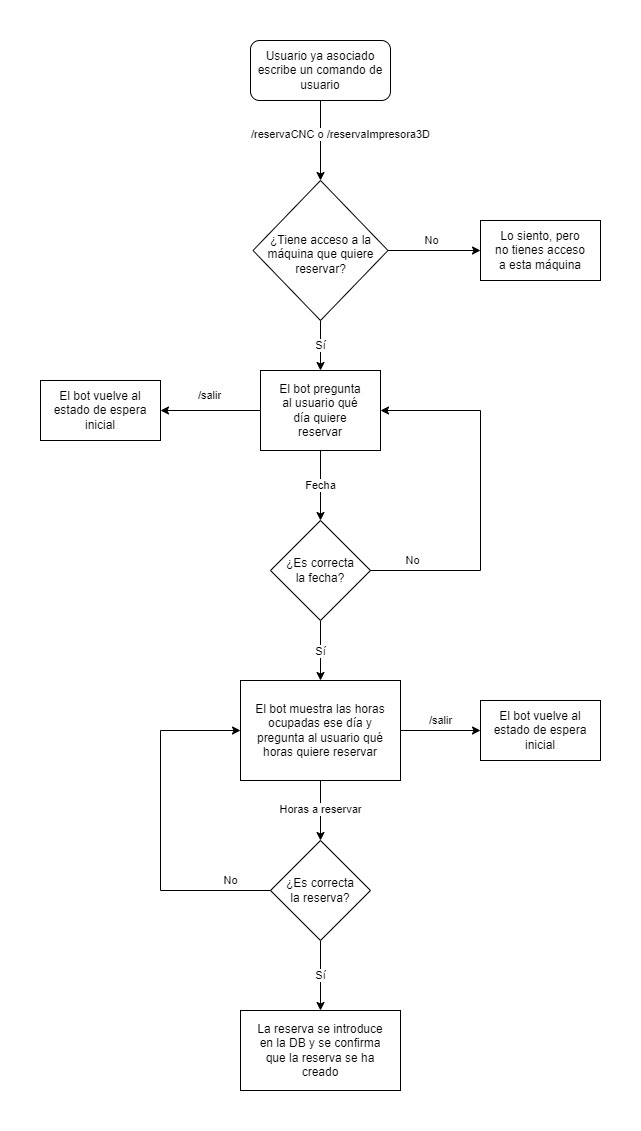
El funcionamiento del bot de Telegram es independiente del proceso de acceso a las salas vía NFC, por lo tanto se pueden mostrar ambos flujos de funcionamiento por separado.

### Bot de Telegram

El siguiente diagrama de flujo muestra el funcionamiento general del bot de Telegram.



Los procesos de reserva de horas para las habitaciones son suficientemente complejos como para merecer un diagrama de flujo independiente.



## Descripción del funcionamiento

### Bot de Telegram

El bot de Telegram “SkapaLaBot” ha sido creado en NodeRed haciendo uso de la librería “node-red-contrib-chatbot” que, entre otras cosas, incluye funcionalidades avanzadas para el diseño de bots de Telegram.

El bot siempre está a la espera de que el usuario escriba algo con el nodo “Telegram Reciever”. Cuando el bot recibe un mensaje, reconoce si el contenido del mensaje es un comando que tiene implementado haciendo uso del nodo “Rules”. Si el mensaje recibido no está dentro de lo que el bot comprende, manda un mensaje al usuario, haciendo uso del nodo “Telegram Sender”, explicándolo e invita a ver el resumen de los comandos implementados haciendo uso de /start.

El comando /start muestra los distintos comandos disponibles. Los comandos están divididos en comandos en comandos públicos y comandos de usuario.

Los comandos públicos son de libre acceso por cualquier usuario que interactúe con el bot. Entre ellos se encuentran:

* **/start** : Muestra un resumen de los comandos disponibles.
* **/contacto** : Muestra información de contacto de SkapaLab.
* **/preguntasFrecuentes** : Muestra preguntas frecuentes sobre SkapaLab.
* **/asociarUsuario** : Inicia el proceso de asociación de id de Telegram con un usuario registrado en la base de datos de SkapaLab.

Para acceder a los comandos de usuario, se comprueba en la base de datos si el id del chat asociado al mensaje que se está recibiendo corresponde a algún usuario registrado en la base de datos. Si existe, se ejecuta el comando. Si no existe, se comienza el proceso de asociación de usuario, también accesible a través del comando /asociarUsuario.

El proceso de registro de un usuario en la base de datos lo hace de manera manual el administrador a través del dashboard. En este registro, entre otra información, se pide una contraseña que se utiliza en el proceso de asociación del usuario.

En el proceso de asociación de usuarios, el bot pide al usuario que escriba su nombre y queda a la espera de la respuesta de este. Se comprueba en la base de datos si el nombre introducido por el usuario se corresponde con el de algún usuario registrado manualmente. Si existe, se pide la contraseña que se proporcionó en el registro físico. Una vez más, se comprueba en la base de datos si el usuario tiene asociada dicha contraseña, si es correcta, el id del chat se almacena en los datos del usuario en la base de datos en el campo “id\_telegram” y se notifica al usuario de que la asociación ha tenido éxito.

Una vez asociado, el usuario tiene acceso a los comandos de usuario:

* **/reservaCNC** : comienza el proceso de reserva de la sala de la CNC.
* **/reservaImpresora3D** : comienza el proceso de reserva de la sala de la impresora 3D.

El proceso de reserva es totalmente análogo para ambas salas, excepto que se guardan por separado las reservas.

Cuando un usuario registrado y asociado introduce cualquiera de los comandos de reserva, se comprueba en la base de datos si, con la suscripción que tiene el usuario, tiene acceso a la sala correspondiente. Si así es, el bot pide al usuario que introduzca la fecha que quiere reservar con el formato “dd/mm/aaaa”, y queda a la espera de la respuesta del usuario. Tras recibir la fecha, se comprueba que sea una fecha válida. Se comprueba el formato, que la fecha exista en el calendario y que sea igual o posterior a la fecha actual. Si se cumplen los requisitos, se muestran las horas disponibles para reserva. Se muestran las horas desde la apertura hasta el cierre y con paréntesis se indican las horas que están ocupadas. Tras eso, el bot indica al usuario que escoja las horas a reservar, mostrando un ejemplo para demostrar el formato. Si el formato es correcto y la franja horaria introducida no está ocupada, la reserva se guarda en la base de datos y se confirma al usuario de que se ha guardado.

Para evitar que un usuario se quede atrapado en un bucle de proceso, se ha implementado el comando /salir, que permite interrumpir cualquier proceso y volver al estado inicial del bot en el que muestra los comandos disponibles.

En varias ocasiones, en los procesos de asociación de usuario y de reserva, se hace uso de variables globales. Esto es debido a que a pesar de que el bloque “Telegram Sender” permita continuar una conversación encadenando mensajes, el mensaje que llega al bloque desaparece y es reemplazado por un mensaje nuevo que contiene la información recibida del usuario. Para conservar esa información a lo largo del flujo de la conversación ha sido necesario usar variables globales como “dia”, “sala” o “nombre”.

## Robustez

En el bot de telegram existen varias comprobaciones en cada paso de la reserva que garantizan que no hayan reservas erróneas. Al ser esta la única manera de reservar, aseguramos que no haya posibilidad de solape, duplicación o formato erróneo en las reservas de la base de datos.

Haciendo uso de la verificación de identidad a través del uso de la contraseña proporcionada en el registro físico, aseguramos que sólo el usuario registrado correspondiente pueda asociar su id de Telegram.

## Posibles mejoras y trabajo futuro

Posibles mejoras que se podrían implementar en el bot serían: un mejor sistema de visualización de las reservas integrando google calendar, más comandos disponibles, cambiar el modo de interacción de comandos por botones y menús, y habilitar la reserva durante la noche para impresoras 3D.

## Librerías

Las librerías usadas en el bot de Telegram son las siguientes:

* **node-red**: librería básica de NodeRed.
* **node-red-contrib-chatbot**: librería para crear chatbots para Telegram, Facebook Messenger, Viber, Twilio y Slack, con gran variedad de funcionalidades.
* **node-red-node-mongodb**: librería usada para para la base de datos.

# Resultados y conclusiones

# Manual de usuario

## Manual del administrador de SkapaLab

Para que el bot funcione con NodeRed es importante asegurarse de que los nodos “Telegram Receiver” y “Telegram Sender” tengan la configuración correcta. Para asegurarse de ello, se hace doble click sobre el nodo a comprobar y haciendo click sobre el símbolo de editar a la derecha del primer desplegable que debe tener seleccionado SkapaLaBot. Debe aparecer en Bot Name “SkapaLaBot”, en Token “5084907312:AAFDpRGPmrdDihcbN7ZjePncQ779rPZBYWc”, en Connect Mode debe estar seleccionado “Polling”, y en Polling debe haber configurado un polling de 1 ms.

Por otro lado, para asegurar que la base de datos y las conexiones MQTT funcionan, también es necesario comprobar que la configuración y autenticación son correctas. Para ello, en ambas usamos como usuario “II8” y como contraseña “kS0Oooyj”.

## Manual del usuario miembro de SkapaLab

Para obtener información de contacto y acceder a preguntas frecuentes sobre SkapaLab basta con comunicarse con el bot de Telegram de SkapaLab. Para ello, en el buscador de Telegram se introduce @SkapaLaBot para encontrarlo por primera vez. Una vez encontrado el bot, mandando el mensaje “/start” recibirás un resumen de los comandos que el bot entiende.

Para poder reservar franjas horarias para usar las salas de CNC e Impresión 3D, necesitas estar registrado en la base de datos con una suscripción. Para ello, es necesario realizar el registro con el administrador. Tras completar el registro se te dará una tarjeta NFC. Una vez estés registrado, en Telegram, usando el comando /asociarUsuario y siguiendo las indicaciones del bot podrás asociar tu usuario con tu cuenta de Telegram. Una vez hecho esto, el bot te reconocerá como usuario registrado y tendrás acceso a los comandos de reserva.

Para acceder a una sala es necesario tener una reserva hecha a través del bot. Puedes hacerlas escribiendo /reservaCNC o /reservaImpresora3D, y siguiendo las indicaciones del bot. Una vez tengas la reserva, en la franja horaria reservada, podrás acceder a la sala seleccionada haciendo uso de la tarjeta que se te proporcionó en el registro con el administrador. Tan sólo es necesario acercarla al lector y una luz verde indicará que se ha abierto.

# Ficheros entregados

* **SkapaLab\_NodeRed.json** : fichero en formato json que contiene los flujos de NodeRed del bot de Telegram “SkapaLaBot”, así como la infraestructura de la base de datos y el dashboard.