



Instituto Tecnológico de Buenos Aires

TP 2

Grupo: UZR

Integrantes:

- **Ponce, Camila Mariana - 59220**
- **Baiges, Matias Sebastian - 59076**
- **Margossian, Gabriel Viken - 59130**
- **Bilevich, Andres Leonardo - 59108**

Índice:

1) Introducción.	3
2) Diferencias y/o similitudes con los prototipos de la primera entrega.	3
3) Decisiones de usabilidad.	6
4) Instructivo de instalación.	12
5) Navegadores soportados.	13
6) Screenshots del sitio web.	13
7) Problemas durante el desarrollo.	27
8) Conclusión.	28

1) Introducción.

En este informe se mostrarán los detalles de la implementación de nuestro sitio Web. Se dispondrán capturas de pantalla y se mostrarán comparaciones y cambios respecto de los prototipos presentados en la primera entrega. Además se listarán las decisiones de usabilidad y se mostrará el conocimiento adquirido a lo largo de la cursada. Por otro lado, se podrán ver datos más técnicos como las instrucciones de instalación y navegadores soportados. Aclaramos que estamos usando la segunda versión del api publicada.

2) Diferencias y/o similitudes con los prototipos de la primera entrega.

Decidimos hacer las cards de los dispositivos (imágenes 2 y 3) y las rutinas (imágenes 1 y 4) más grandes. Aunque utilizando el diseño anterior hubiéramos podido incluir más cantidad de ellas por pantalla, y evitar mucho scrolleo, al ser pequeñas podía ser incómodo para personas con monitores no muy grandes y los botones serían más difíciles de clicar con el mouse. De esta forma, también generamos más espacio para mostrar el status de cada dispositivo mientras la tarjeta está cerrada, cosa que no se podía hacer con el anterior diseño por falta de espacio para algunos dispositivos en específico, como los parlantes o el horno. Al hacer esto, bajamos los clicks y al recordarle nosotros al usuario el estado del dispositivo hacemos que piense menos.

Prototipo

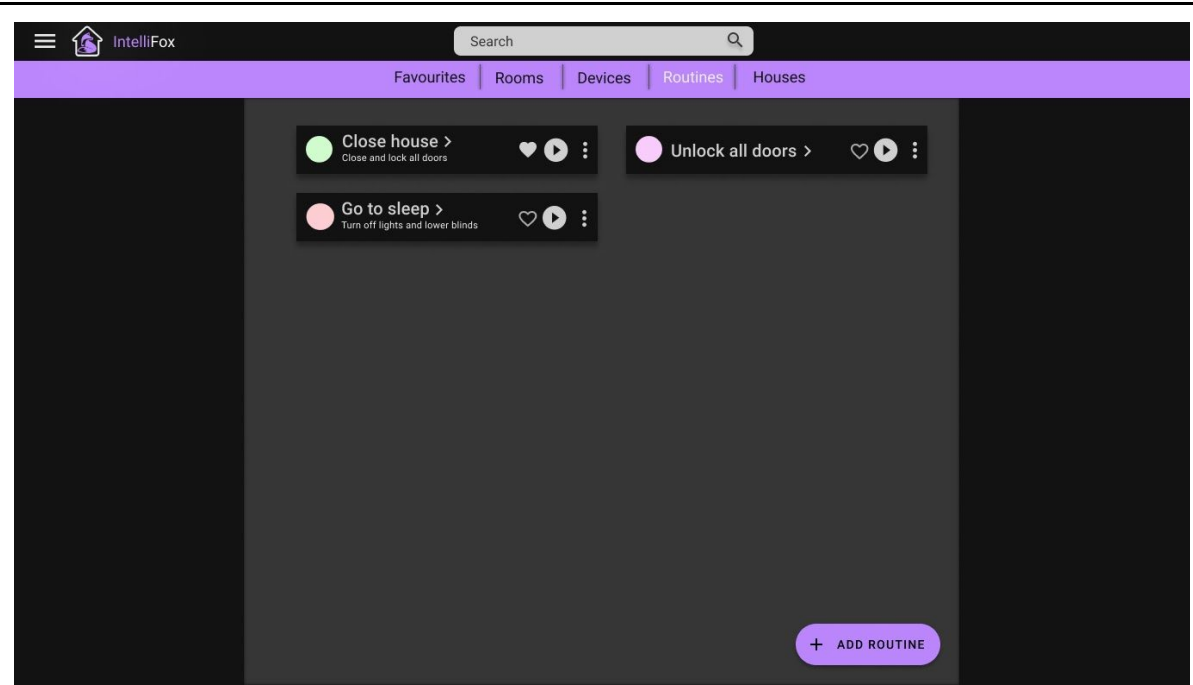


imagen 1

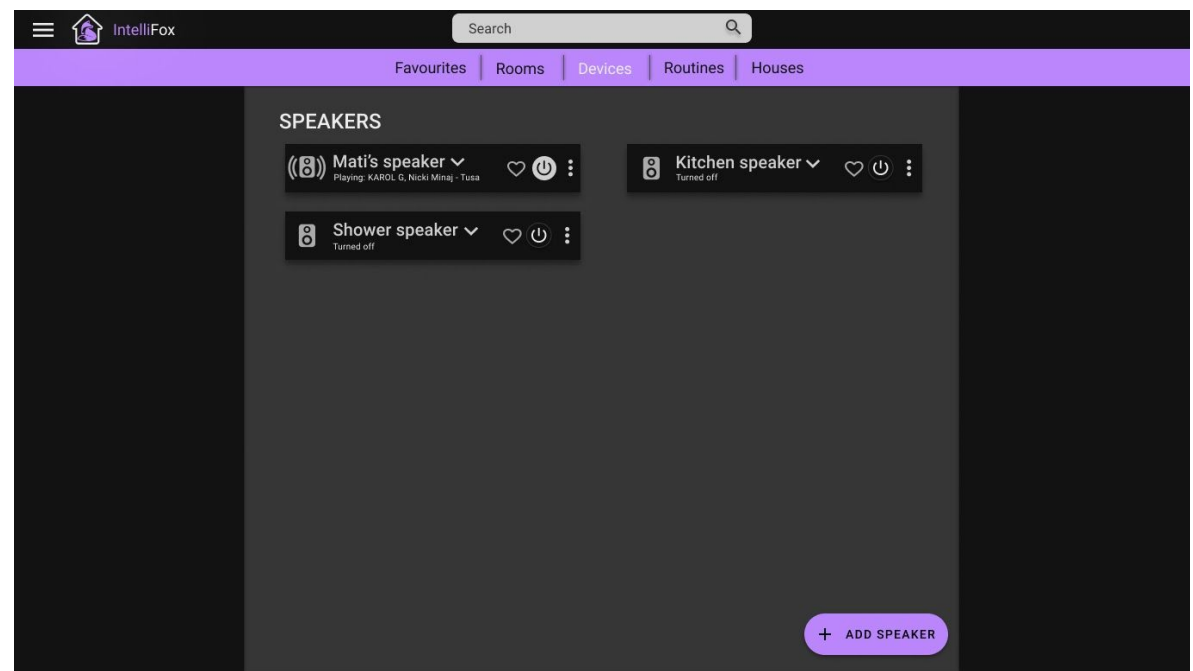


imagen 2

Implementación

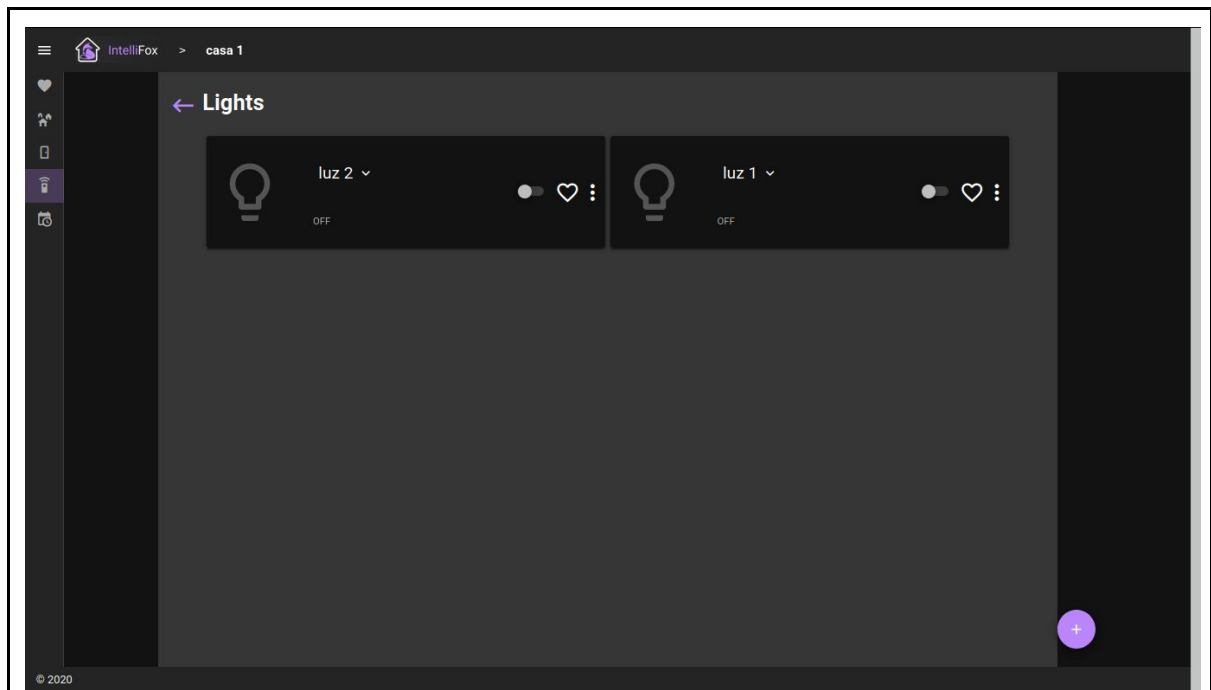


imagen 3

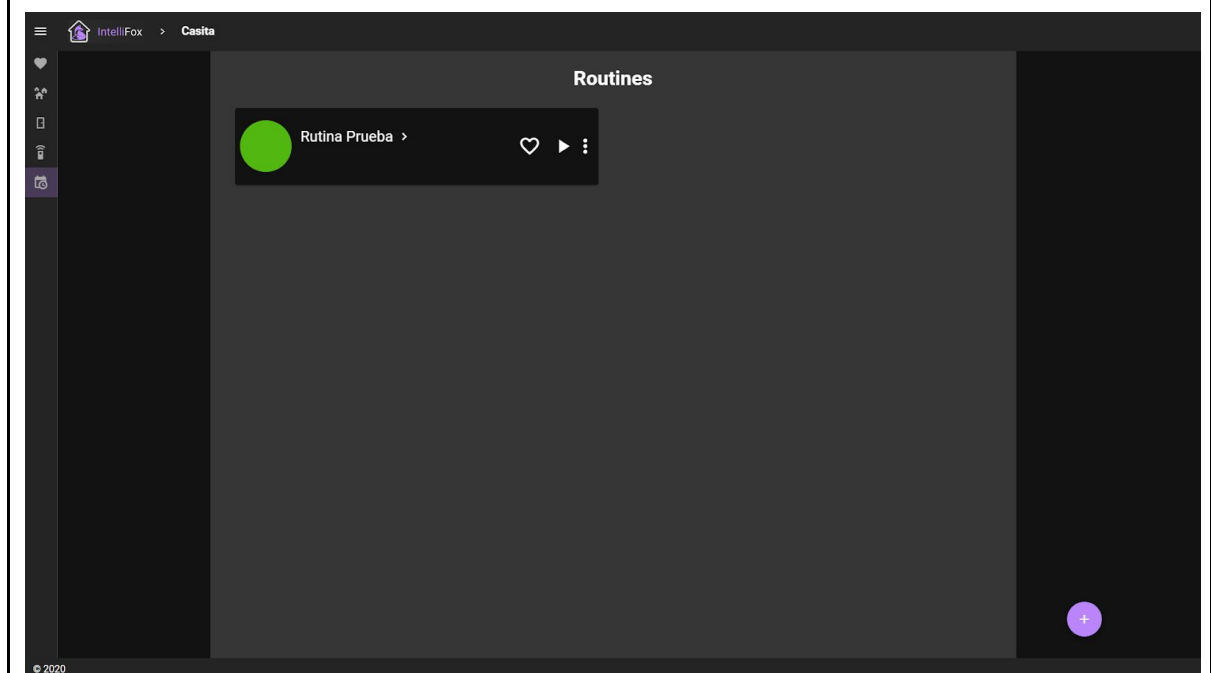


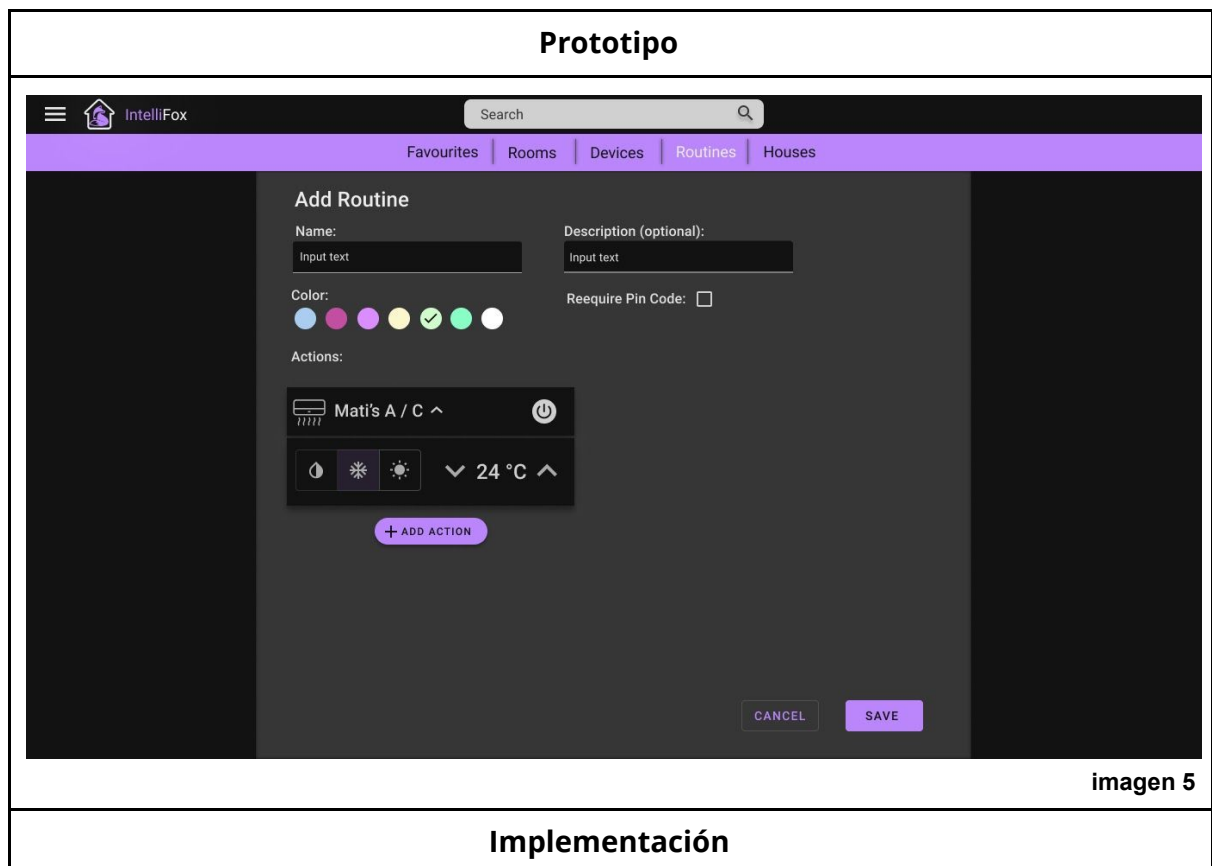
imagen 4

Quitamos la descripción de los dispositivos ya que coincidimos todos en que no aportaba nada y podría hasta incluso generar ruido visual. Decidimos que el nombre ya era suficiente para diferenciarlo.

Cambiamos el botón de encendido de los dispositivos por un switch ya que para algunos dispositivos como por ejemplo las persianas, el botón anterior generaba algo de confusión sobre en qué estado se encontraba y que acción se realizaba al presionar

el mismo. Con el switch queda más claro y coherente. Aun así, también se muestra el estado actual del dispositivo como descripción en todo momento.

Además, decidimos incorporar más colores a la elección de rutinas (imagen 6), para que el usuario tenga más opciones y para evitar posibles confusiones de usuarios con dificultades en la vista. En vez de una selección limitada de colores como se había propuesto en el prototipo (imagen 5) se implementó un selector de colores que permite escoger entre una amplia selección de tonos.



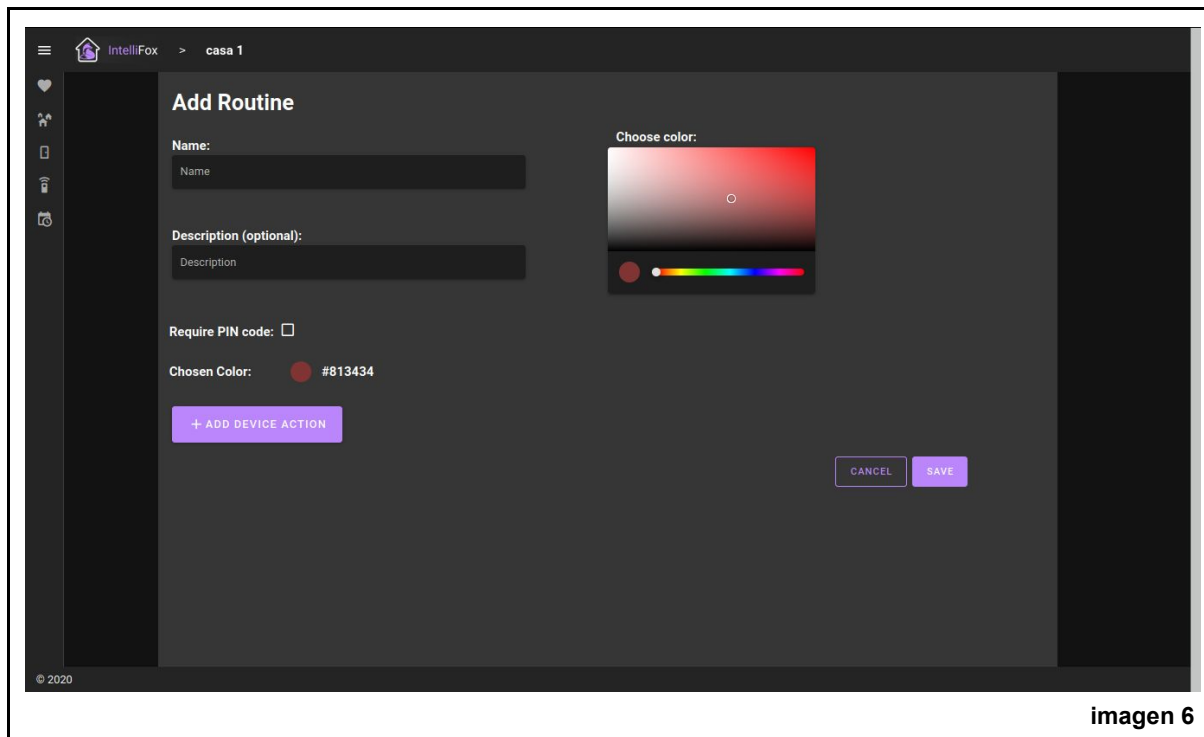


imagen 6

Modificamos el color de algunos iconos y botones como los de borrado ya que eran rojos. Buscamos evitar estos tonos ya que suelen estar asociados a errores.

3) Decisiones de usabilidad.

Incluimos un título para cada pantalla para ayudar al usuario a saber donde se encuentra y hacer que piense menos. Además, en caso de que alguna página no tenga aún ningún ítem incorporado, se le muestra al usuario un mensaje indicando la situación, y recordando que puede agregar un nuevo ítem, dando una breve instrucción de cómo hacerlo.

Decidimos cambiar la forma de navegación principal. Quitamos el menú horizontal de la franja violeta (imagen 7) y se implementó un menú vertical, debajo del navigation drawer. Esto lo hicimos para tener una sola forma de navegación en vez de dos ya que esto podría traer confusión a algunos usuarios. En la pantalla siempre se muestran los iconos de nuestras 4 opciones principales que son Homes, Rooms, Devices y Routines, junto con la página de favoritos (imagen 8). Al presionar el navigation drawer, se muestran más en detalle y aparecen las opciones de History, Settings, Help & Feedback y Error Log. Esta nueva forma de navegación está inspirada en la que utiliza la empresa Google en la página de Youtube.

Sumado a lo anterior, el navigation drawer de nuestro prototipo abría un menú de overflow (imagen 7), lo que no se alinea al modelo mental de los usuarios. Esto también nos ayudó a tomar la decisión de modificar la página en la implementación.

Prototipo

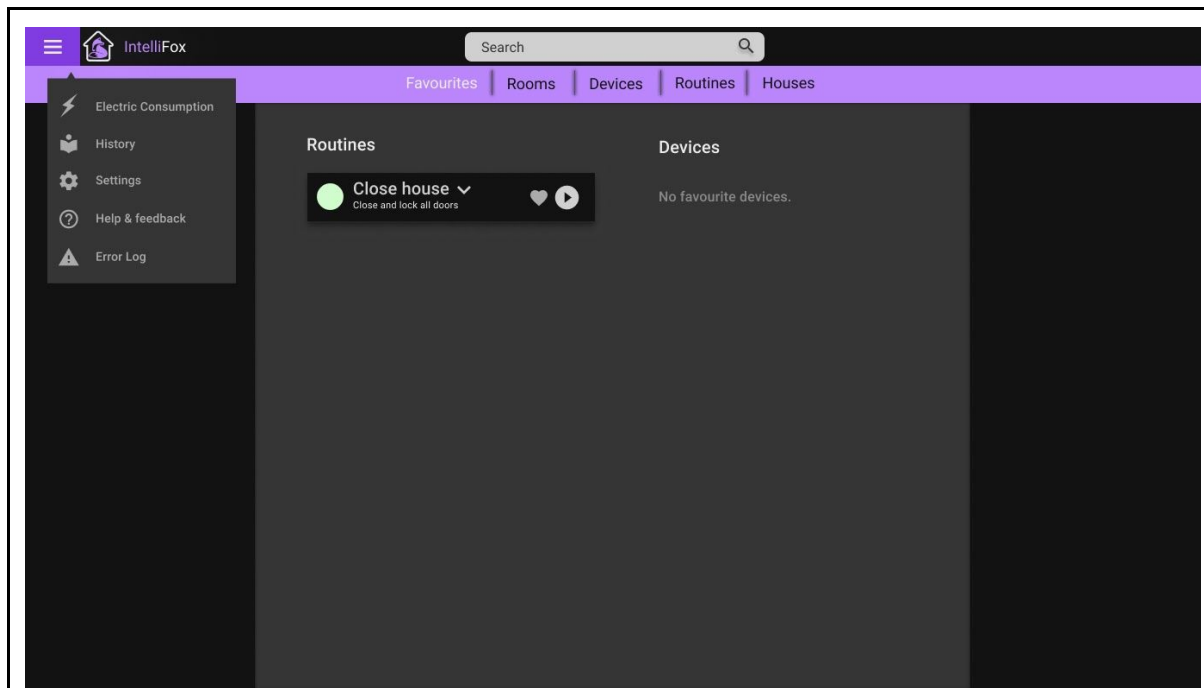


imagen 7

Implementación

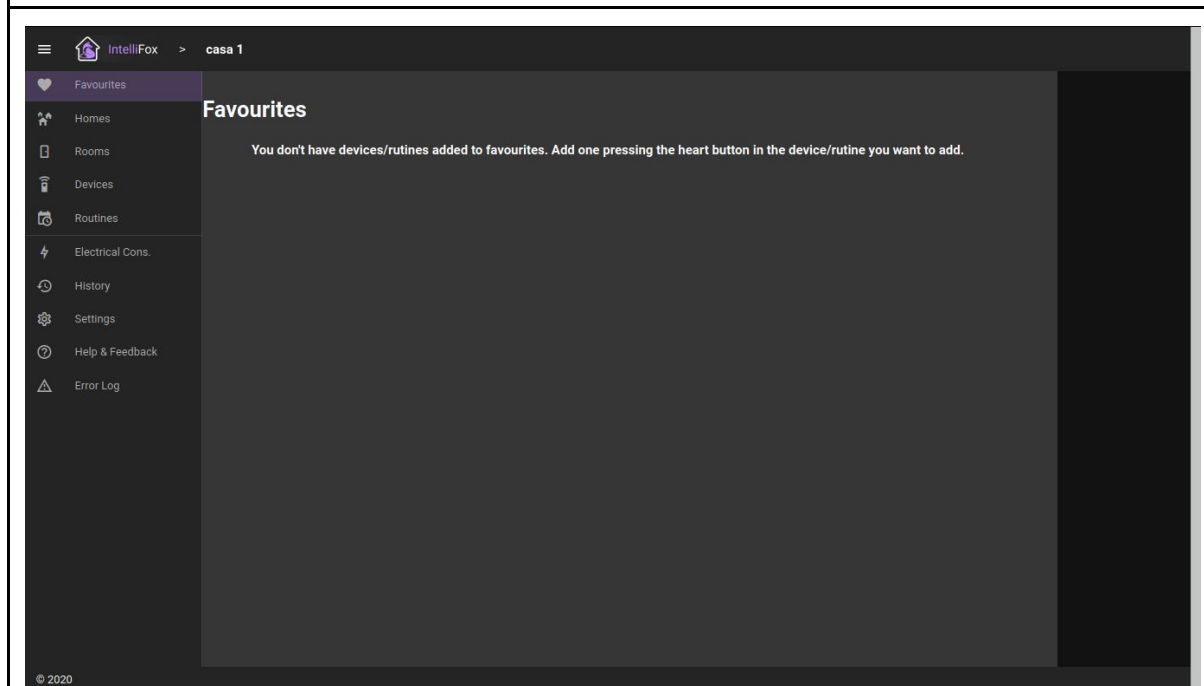


imagen 8

Modificamos la forma en la que se manejan las rutinas. En nuestro prototipo se había propuesto agregar los dispositivos y el estado específico el que tendrían que estar luego de ser ejecutada la rutina, lo que podría ser molesto si se quisiera realizar solo 1 acción del dispositivo o de varios dispositivos (imágenes 9 y 10). Para mejorar esto, a la hora de implementarlo se decidió que primero se agregaría el dispositivo (imagen 11 y 12) y luego las acciones que este realizará por separado (imagen 13 y 14), logrando que la

rutina se vea más organizada y sea más cómoda para manejar y visualizar. Esto ayuda a que el usuario no tenga que recordar y pueda reconocer fácilmente la rutina que desea.

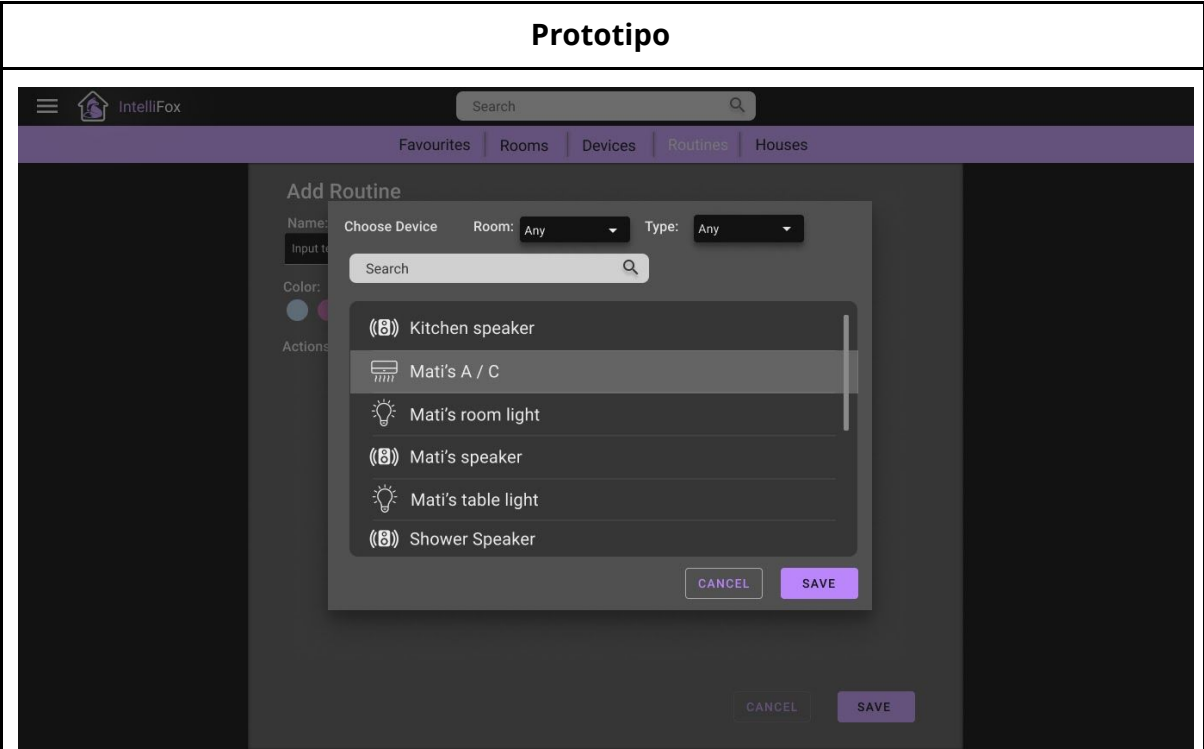


imagen 9

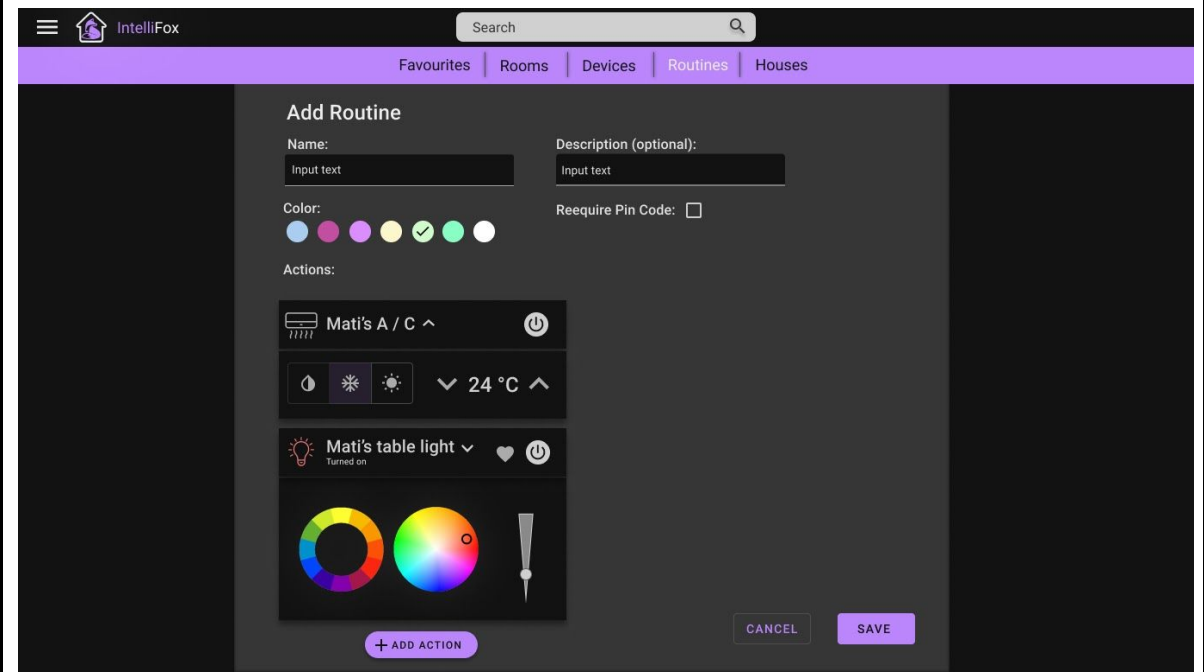


imagen 10

Implementación

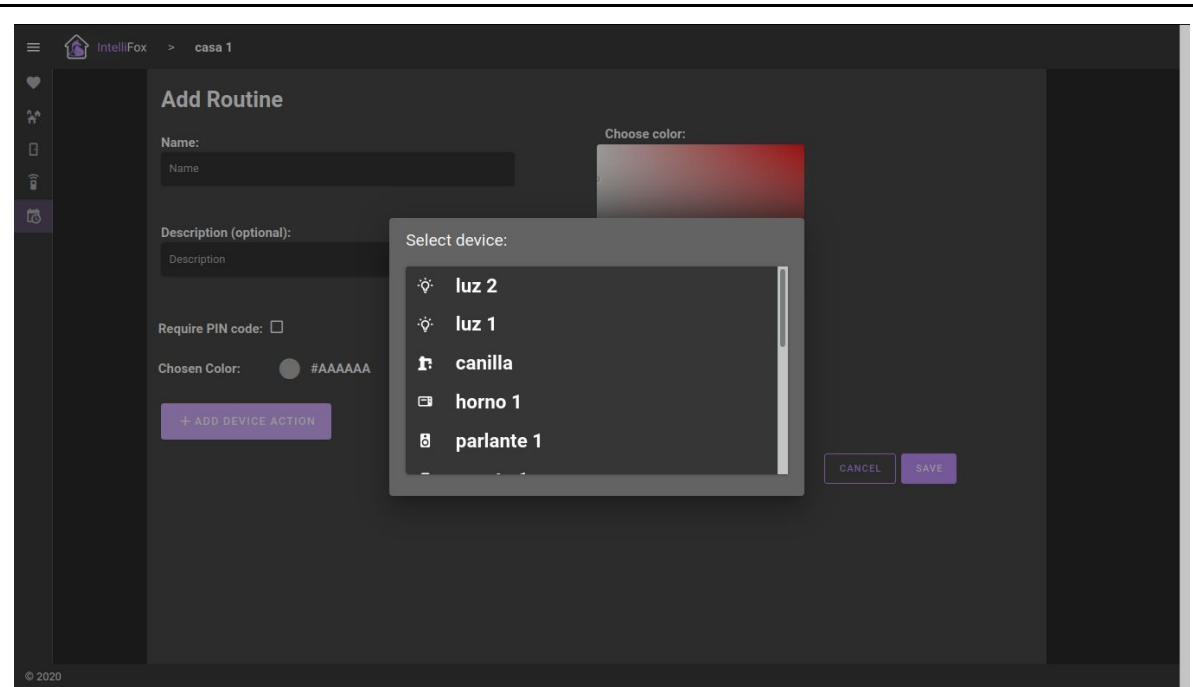


imagen 11

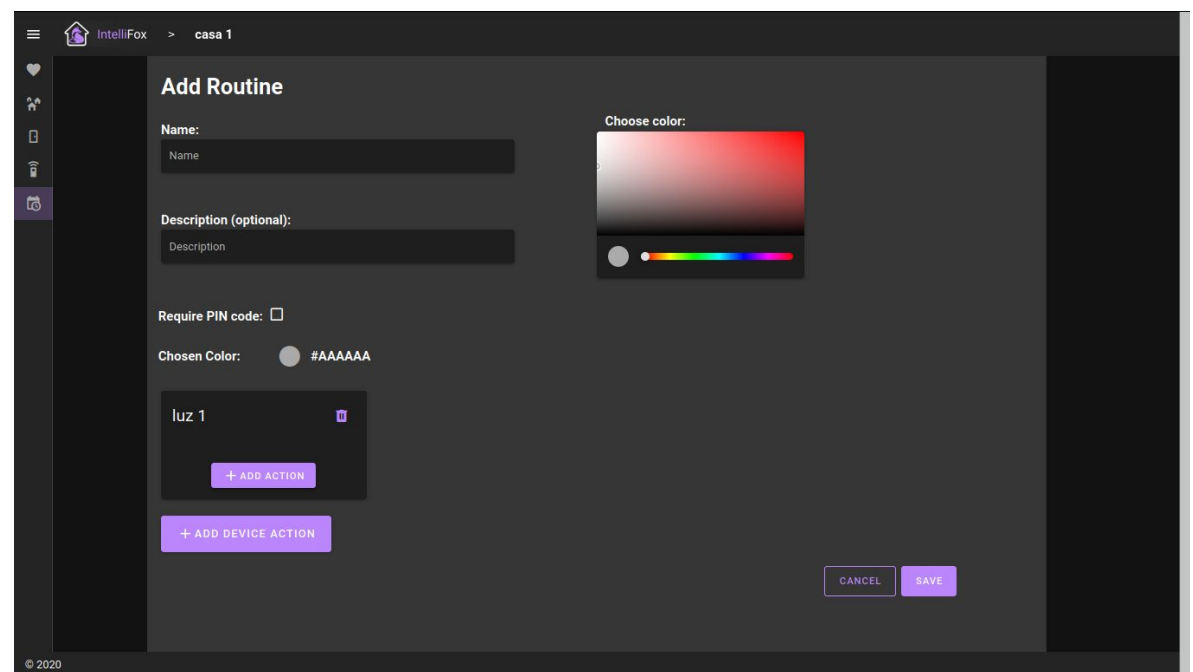


imagen 12

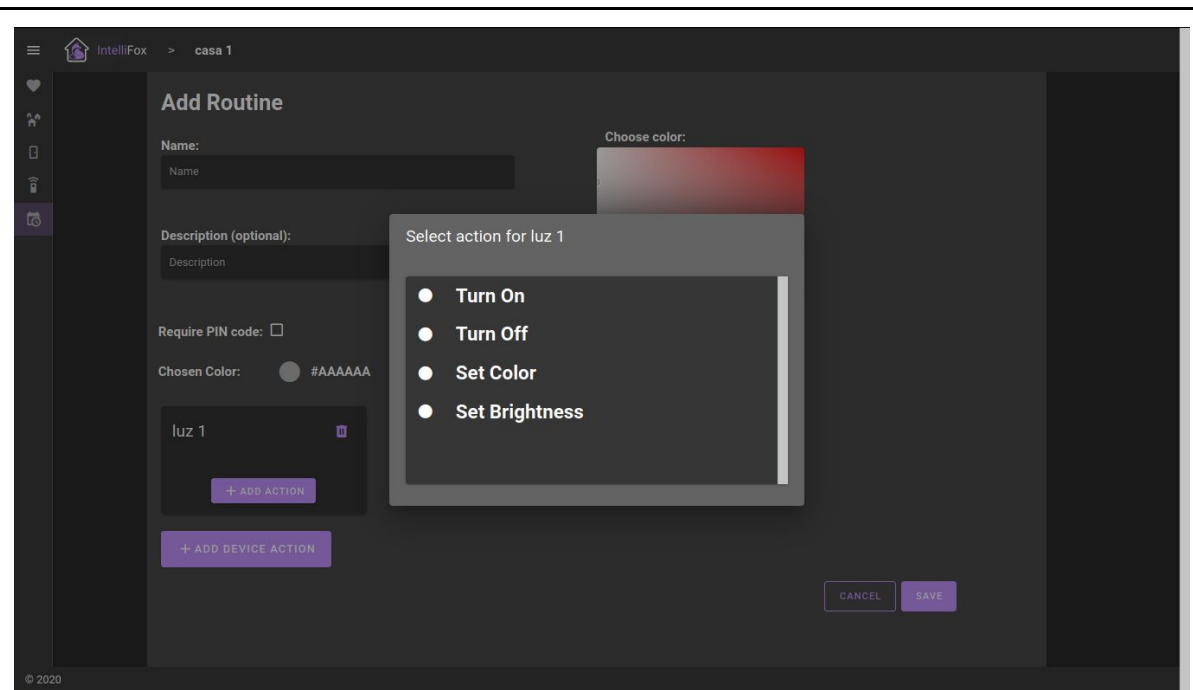


imagen 13

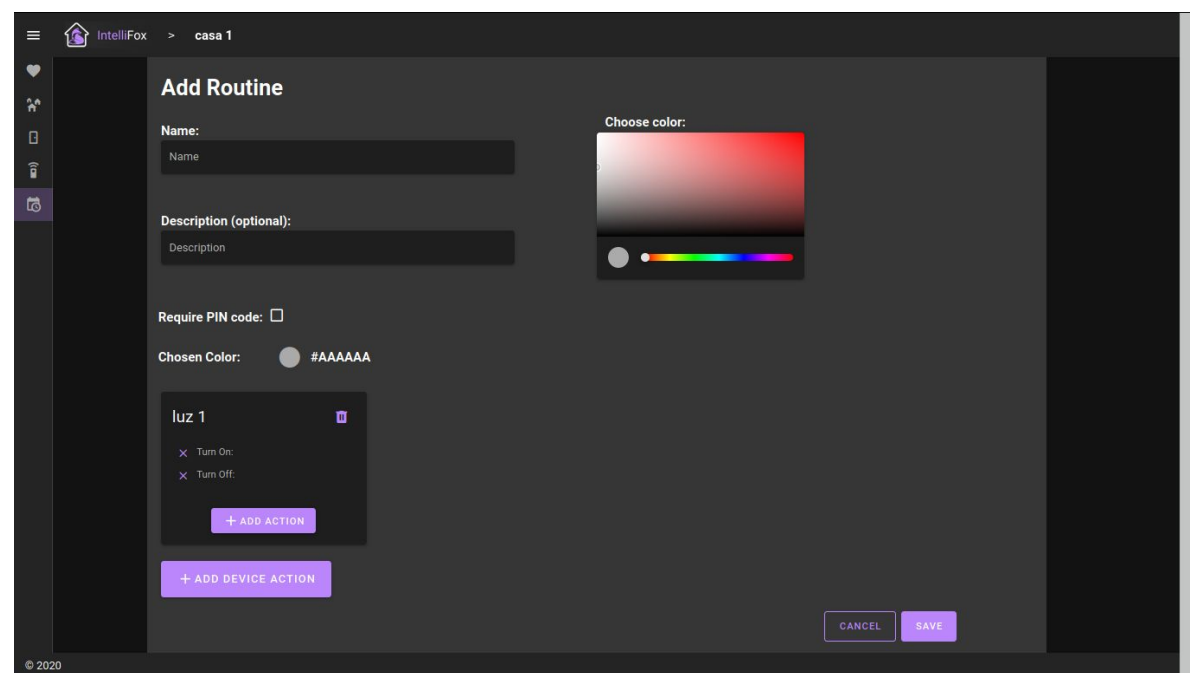


imagen 14

Se incorporaron iconos específicos de material design a lo largo del sitio, maximizando la coherencia con el mundo real y modelos mentales, lo que hace nuestra implementación visual e intuitiva. Además, en ciertas ocasiones cómo la ejecución de acciones en los dispositivos, se dispuso que el switch del dispositivo entre en un modo de “cargando” mientras se cambia el estado para poder así visualizar el estado del sistema. Esto, junto con otros elementos tales como cambios en iconos (como al prender / apagar un dispositivo) o la creación/borrado de una carta (lo que ocurre al agregar / borrar un nuevo cuarto / dispositivo / hogar / rutina), ayudan a informarle al

usuario sobre lo que está ocurriendo y a hacer que este sienta el control sobre lo que hace.

También se hizo énfasis en la prevención de errores, ya que se dispusieron alertas antes de realizar un borrado definitivo de algún elemento del sitio, preguntando al usuario si está seguro de querer realizar esta acción.

Se mantuvo además, en líneas generales, el mismo diseño minimalista utilizado en los prototipos presentados en la primera entrega. Se mantuvo una misma paleta de colores a lo largo de la implementación y se evitaron los malos contrastes. Además, nuestro sitio es consistente visualmente en cuanto a la alineación, colores y tamaños de fuente en todos los ámbitos.

Nuestro sitio evita el uso de largos textos para no generar ruido visual. Se maneja principalmente mediante iconos y botones. Esto ayuda también a que el usuario no tenga que ingresar demasiada información, y pueda tener todo a su alcance de manera rápida y sin necesidad de recordar muchos datos.

4) Instructivo de instalación.

Para visualizar correctamente nuestro sitio realizar los siguientes pasos:

Para la API:

1. Descomprimir el archivo "API v1.6.0 Installer.zip" en algún directorio temporal.
2. Descargar de <https://nodejs.org/en/> la versión 10.15.3 o superior de node.js correspondiente a la plataforma.
3. Crear un nuevo directorio donde quedará instalada el API una vez finalizado este instructivo.
4. Copiar el archivo "package.json" que se encuentra en el directorio temporal del paso 1 al directorio creado en el paso 3.
5. Utilizando la consola de línea de comandos o una terminal, posicionarse en el directorio creado en el paso 3, escribir "*npm install*" (sin las comillas) y presionar la tecla Enter.
6. Copiar los subdirectorios actions, config, db, initializers, locales, node_modules y public que se encuentran dentro del directorio temporal del paso 1 al directorio creado en el paso 3. Durante la copia se solicitará la confirmación para sobrescribir archivos existentes, aceptar la misma.
7. Utilizando la consola de línea de comandos o una terminal, posicionarse en el directorio creado en el paso 3, escribir "*.\node_modules\.bin\actionhero link --name=ah-swagger-plugin*" (sin las comillas y ajustando las barras de acuerdo a la plataforma) y presionar la tecla Enter. Ignorar la advertencia que se muestra por la existencia del archivo "swagger.js".
NOTA: en macOS a veces el módulo "action-hero" no se instala correctamente. Si este fuera el caso se obtendrá un error en el paso 7 al ejecutar el comando especificado. Para resolver el problema escribir "*npm install action-hero*" y presionar la tecla Enter. Una vez finalizada la instalación del módulo volver a repetir el paso 7.
8. Correr el comando *npm start* en una terminal situada dentro de la carpeta del paso 3 y cerrarlo.

9. Descomprimir el archivo "API v1.6.1 Installer.zip" en algún directorio temporal.
10. Copiar el archivo "package.json" y el subdirectorio actions que se encuentran en el directorio temporal del paso 9 al directorio de instalación del API del paso 3.
11. Para correr la API, ejecutar el comando `npm start` en el directorio del paso 3.

Para el front:

12. Descomprimir los archivos del zip "HCI_Web.zip" en un directorio.
13. Definir en qué puerto se le asignará al API. Para esto, dentro del directorio `/Intellifox/src/api/` dirigirse al archivo `api.js` y definir el puerto deseado (en nuestro caso el 8080).
14. En una terminal, dentro de la carpeta *Intellifox* ejecutar el comando "`npm install`".
15. Ir a la carpeta donde está instalada la API y dentro de ella, en otra terminal, ejecutar el comando "`npm start`".
16. Volver al directorio anterior y ejecutar en la consola el comando `npm run serve`.
17. Al terminar de efectuarse el comando, copiar y pegar en un navegador web el enlace al localhost que aparecerá en pantalla.

5) Navegadores soportados.

- Chrome 70 o superior soportado.
- Edge 40 o superior soportado.
- Firefox 70 o superior soportado.
- Safari 10 en adelante soportado.

6) Screenshots del sitio web.

En esta sección se incorporan las capturas de pantalla que no se han incluido en secciones anteriores.

1. Homes:

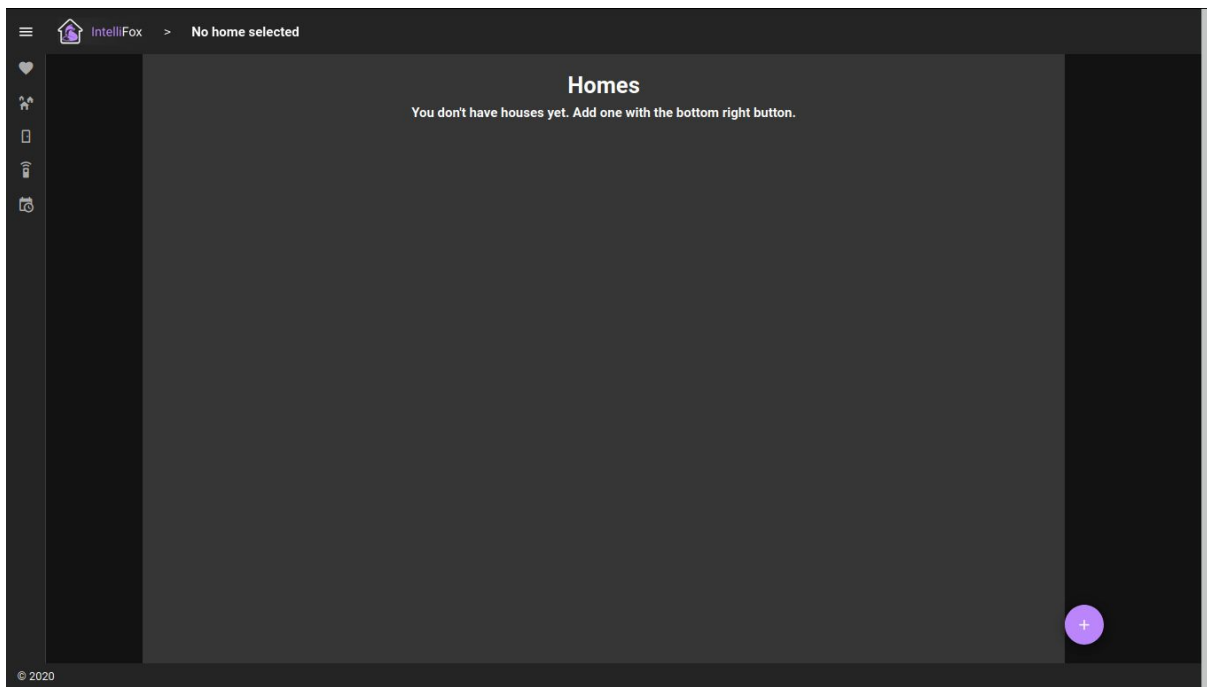


imagen 15 - Hogares (vacío)

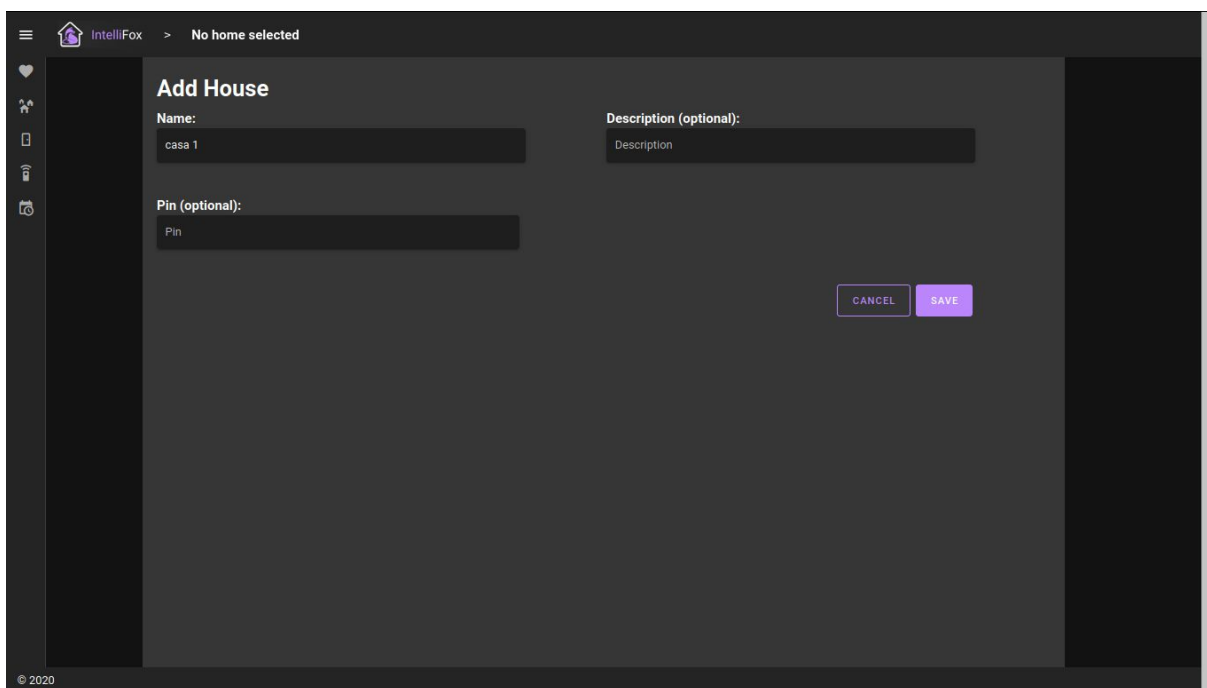


imagen 16 - Agregar hogar

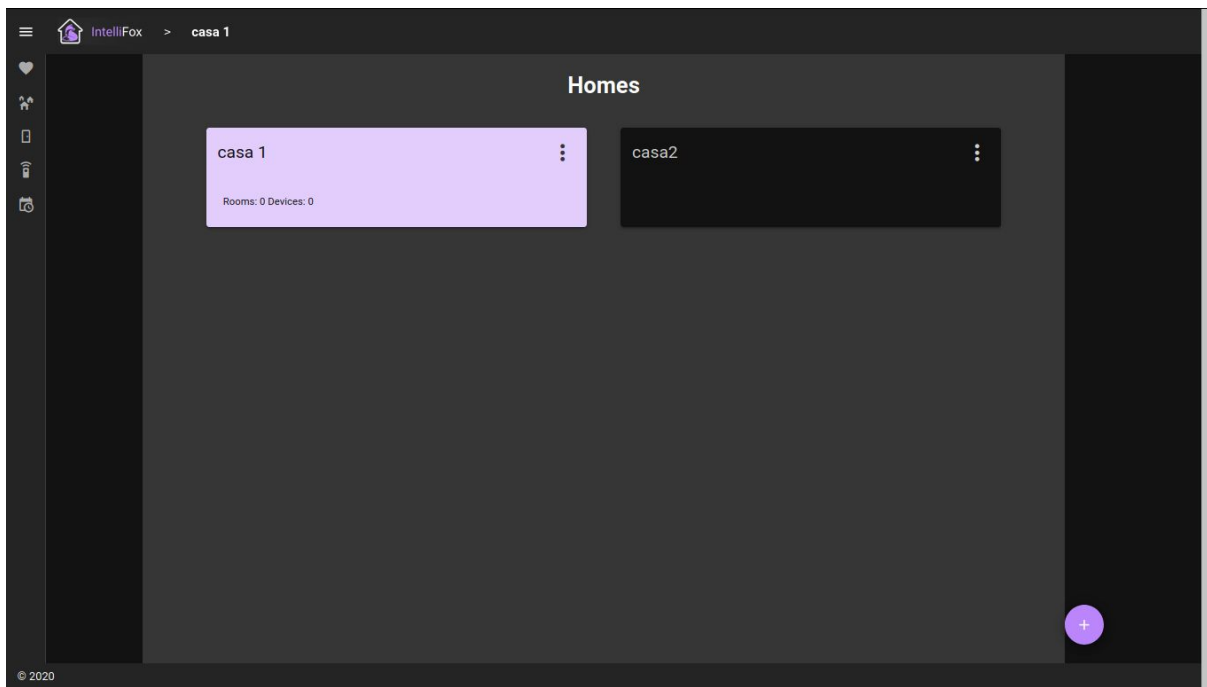


imagen 17 - Hogares

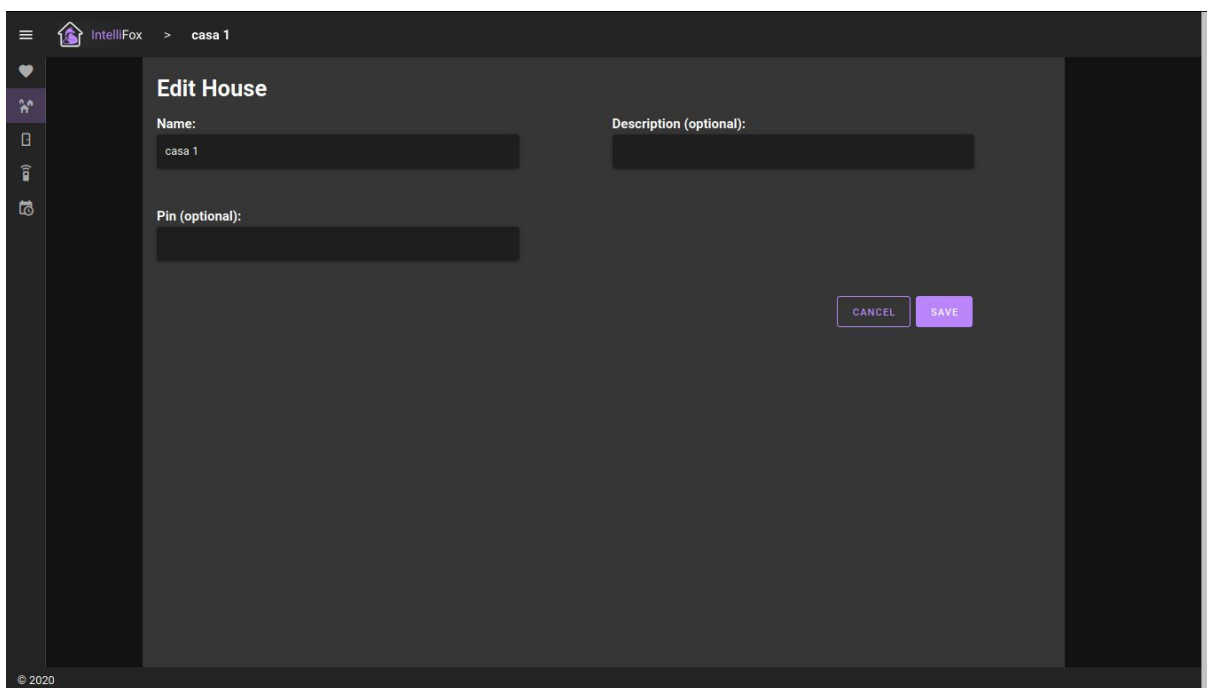


imagen 18 - Editar hogar

2. Rooms:

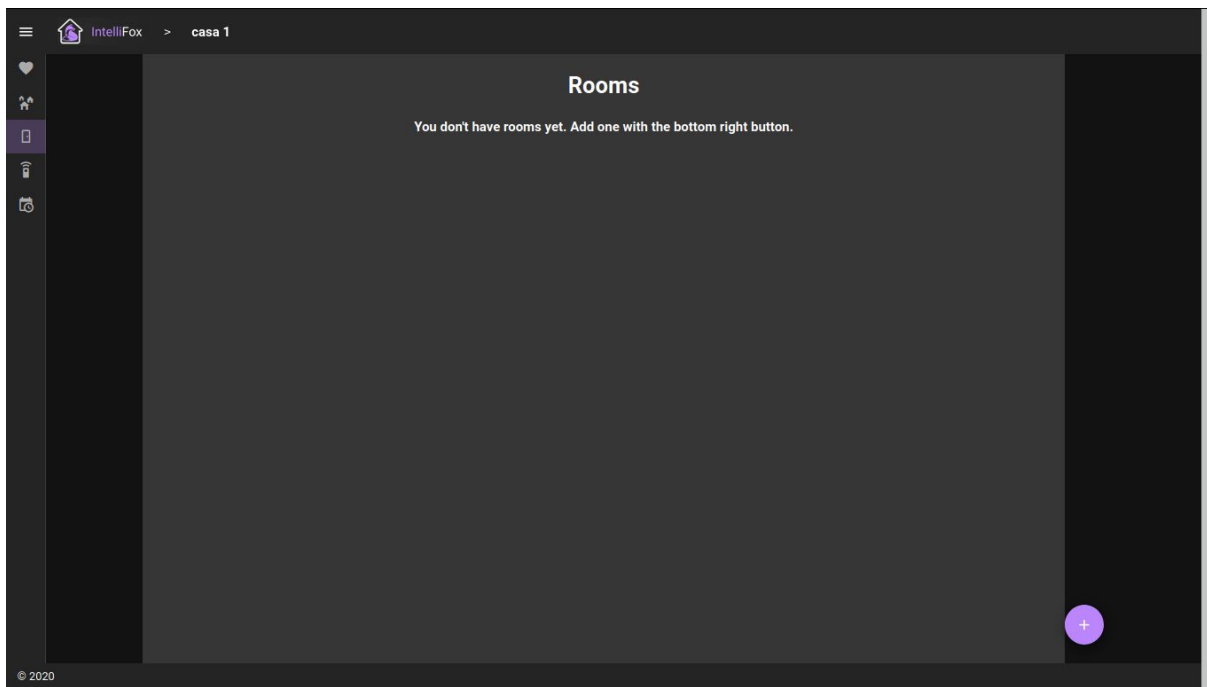


imagen 19 - Habitaciones (vacío)

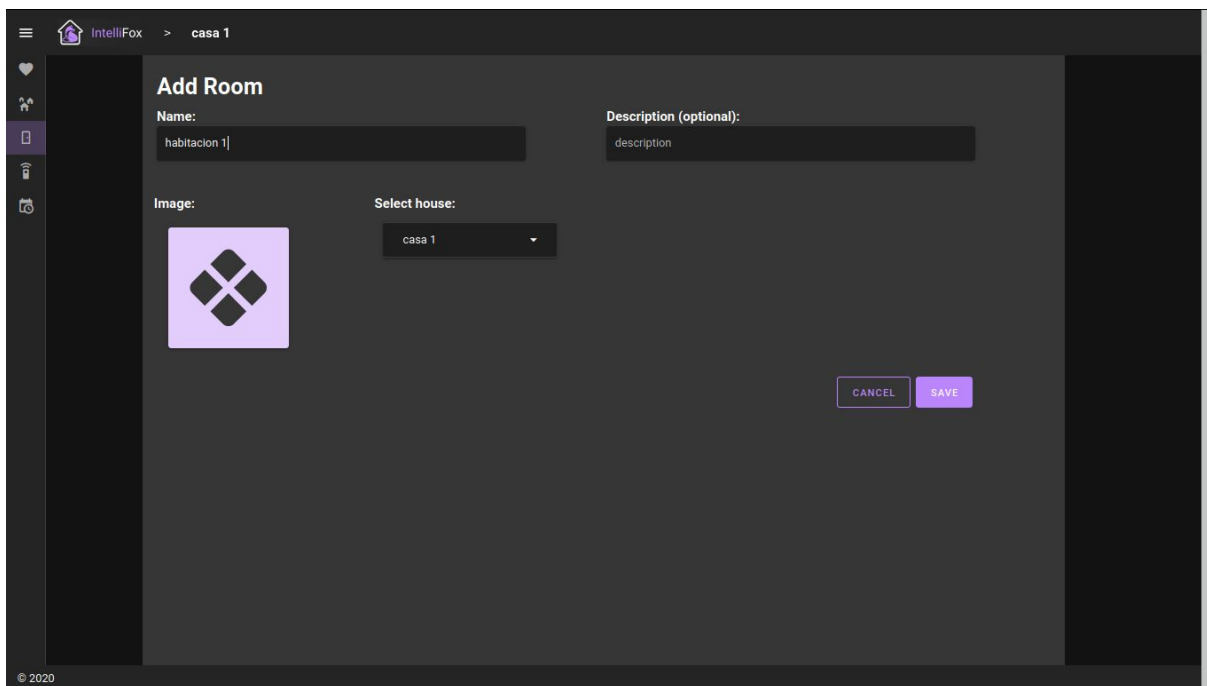


imagen 20 - Añadir habitación

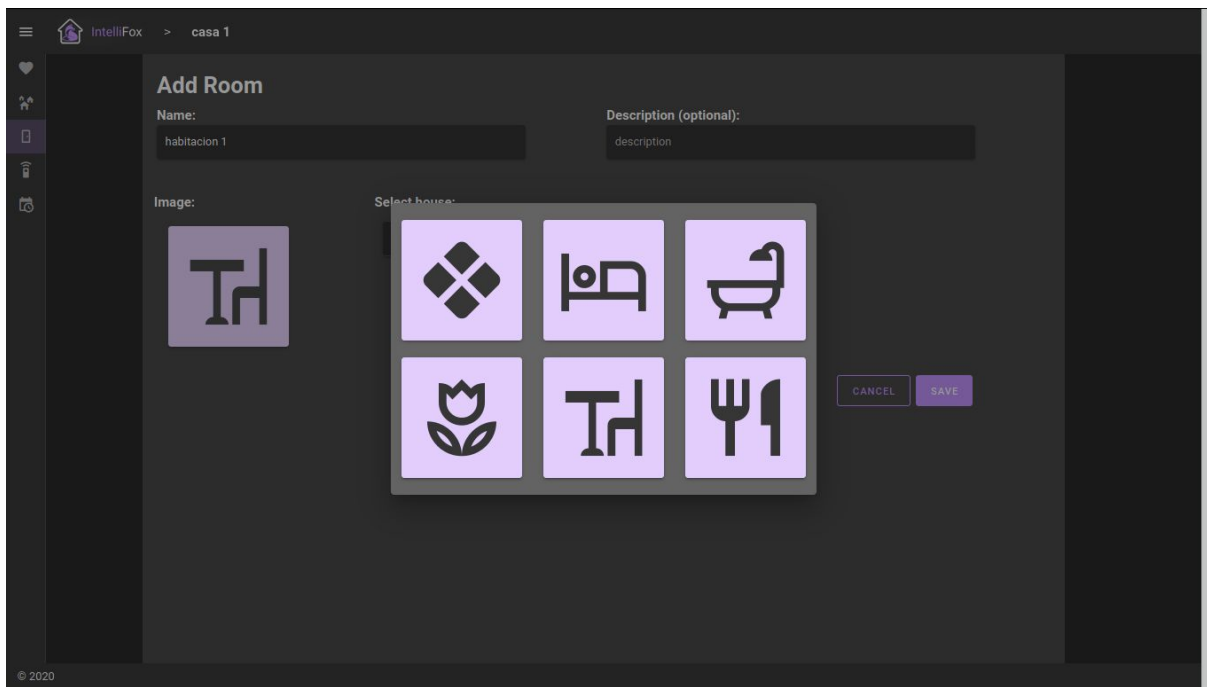


imagen 21 - Añadir habitación (Seleccionar imagen)

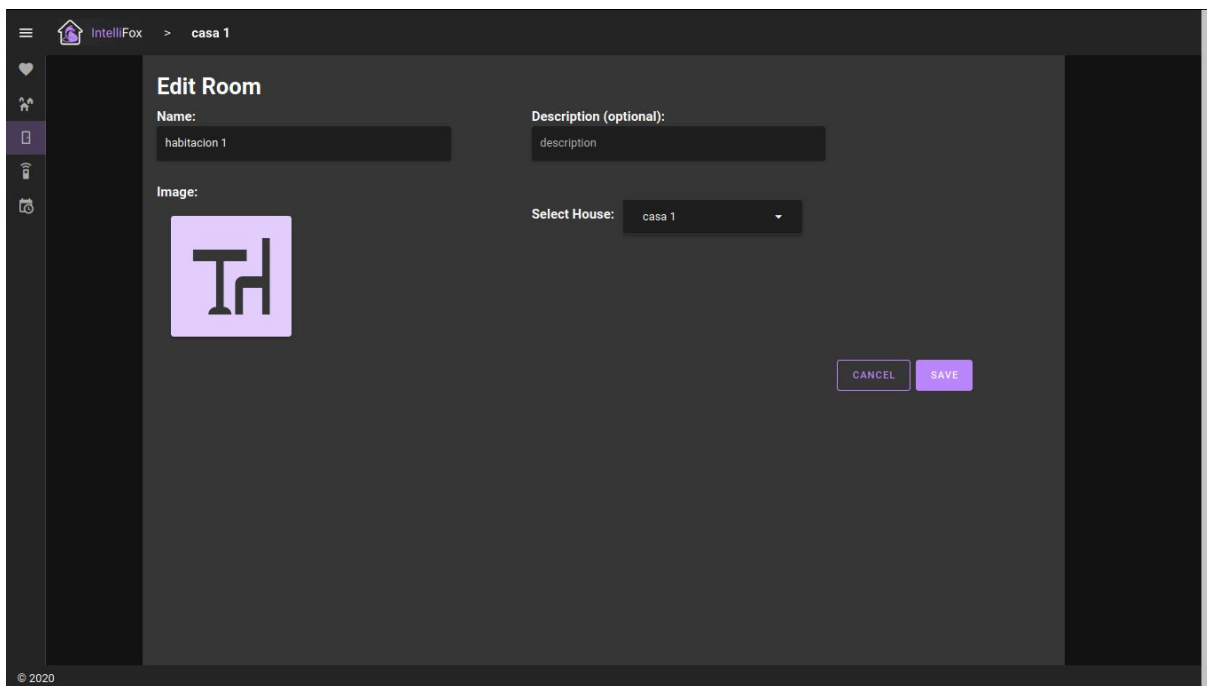


imagen 22 - Editar habitación

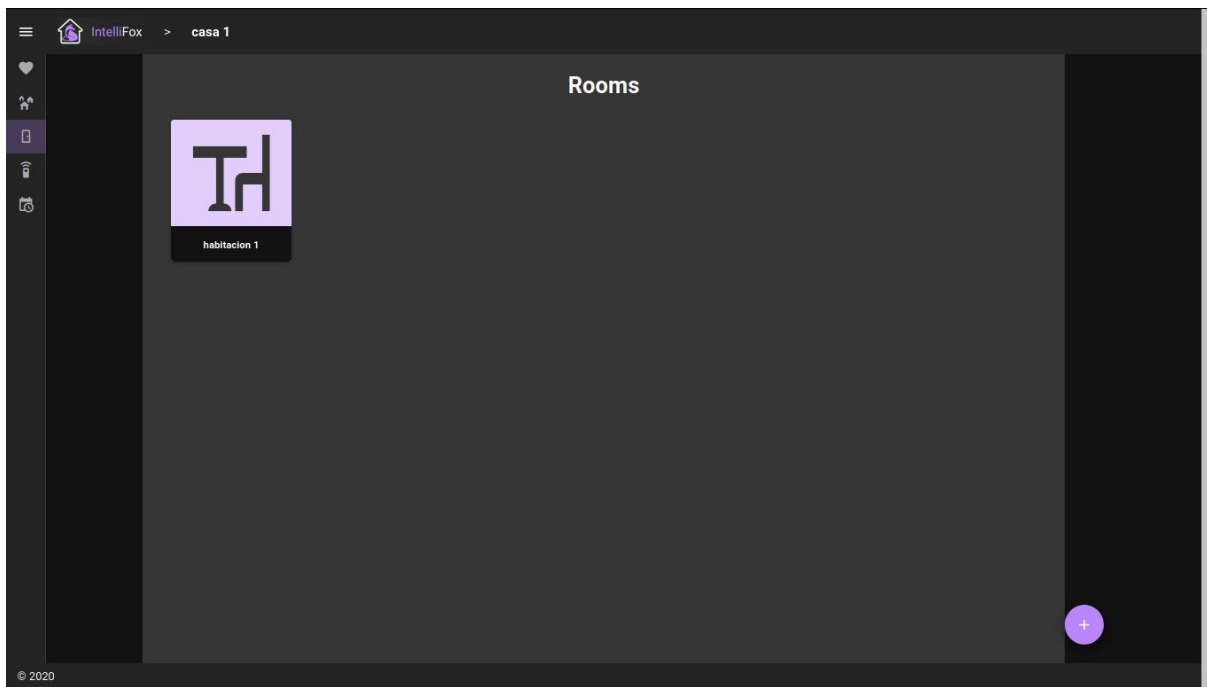


imagen 23 - habitaciones

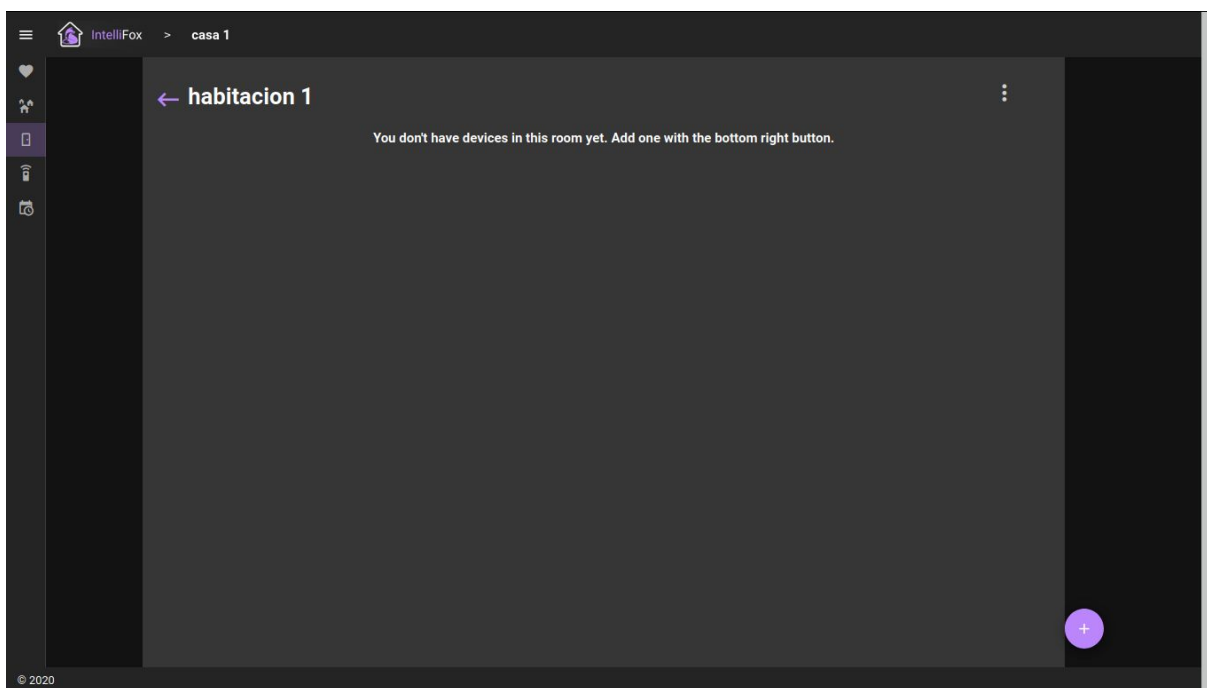


imagen 24 - Dentro de una habitación (vacía)

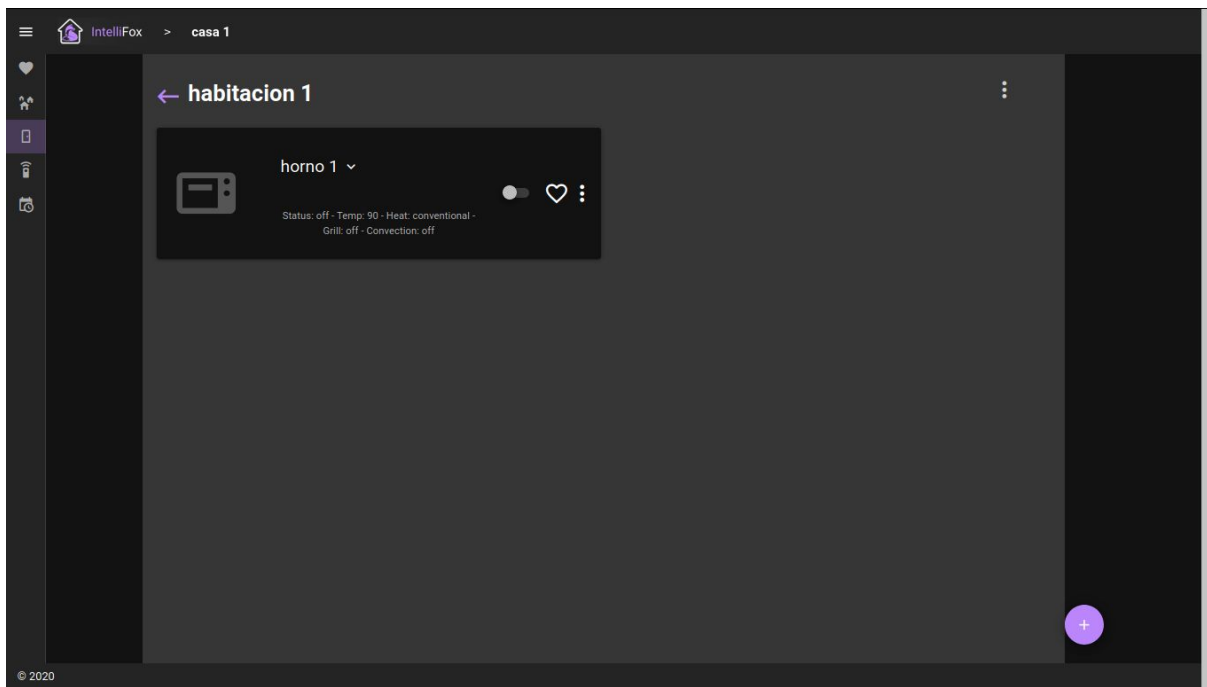


imagen 25 - Dentro de una habitación

3. Devices:

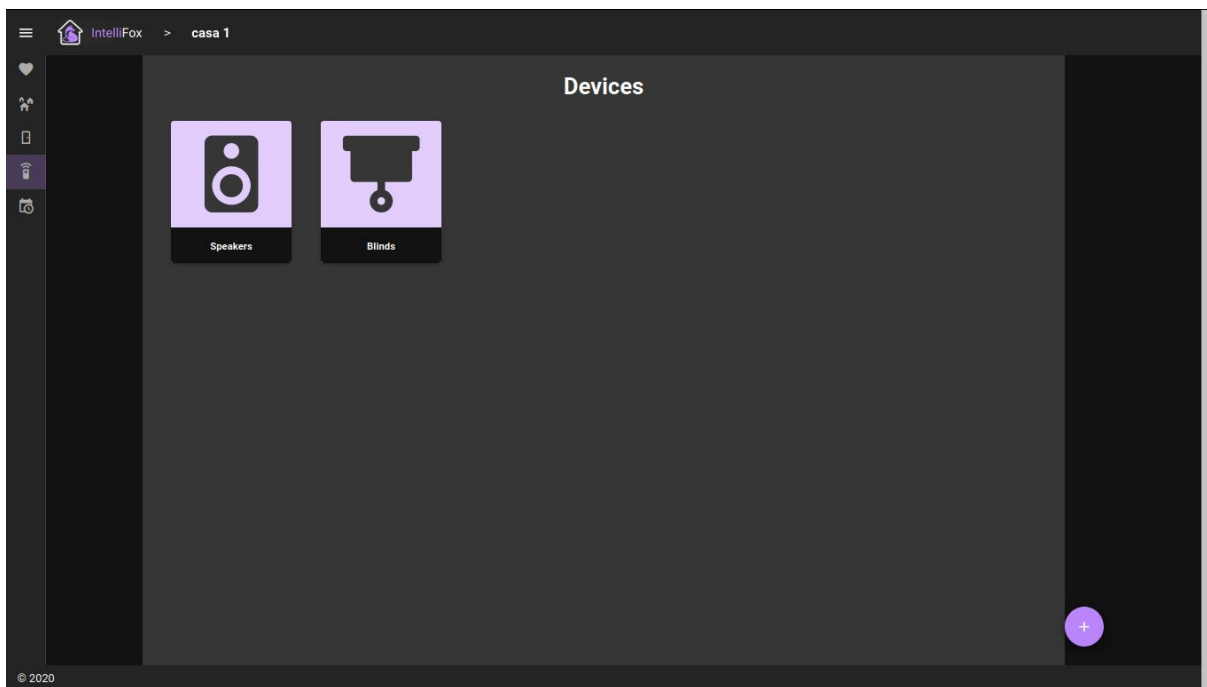


imagen 26 - Dispositivos

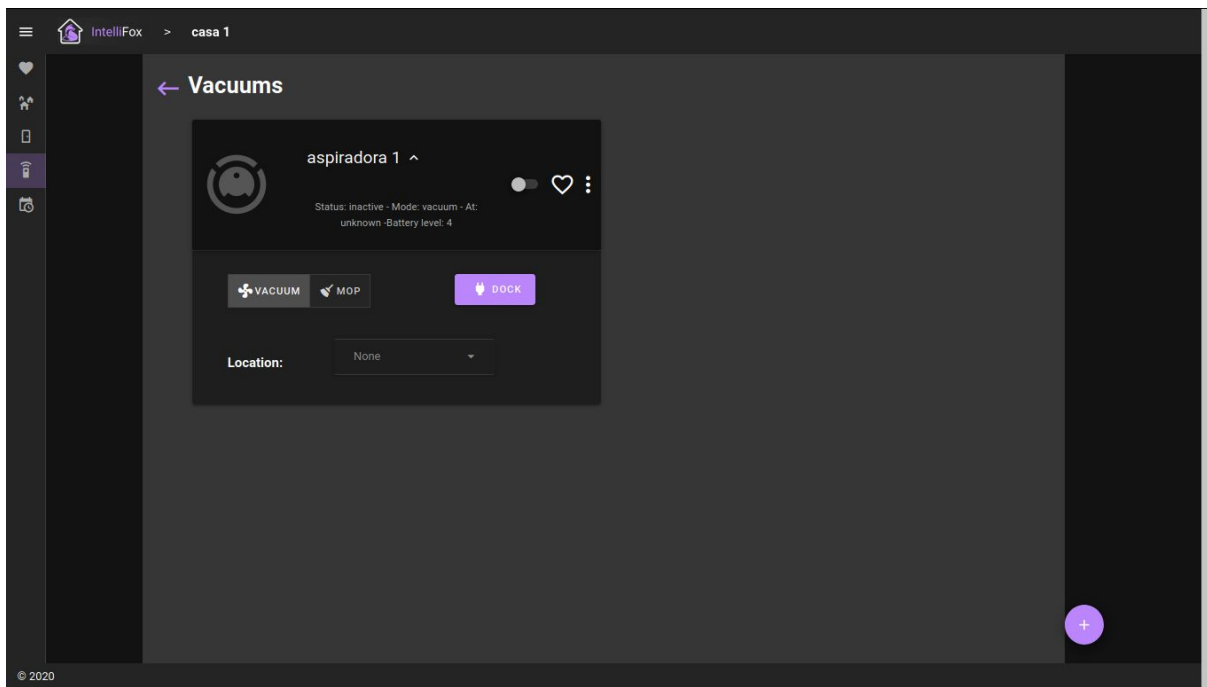


imagen 27 - Aspiradoras

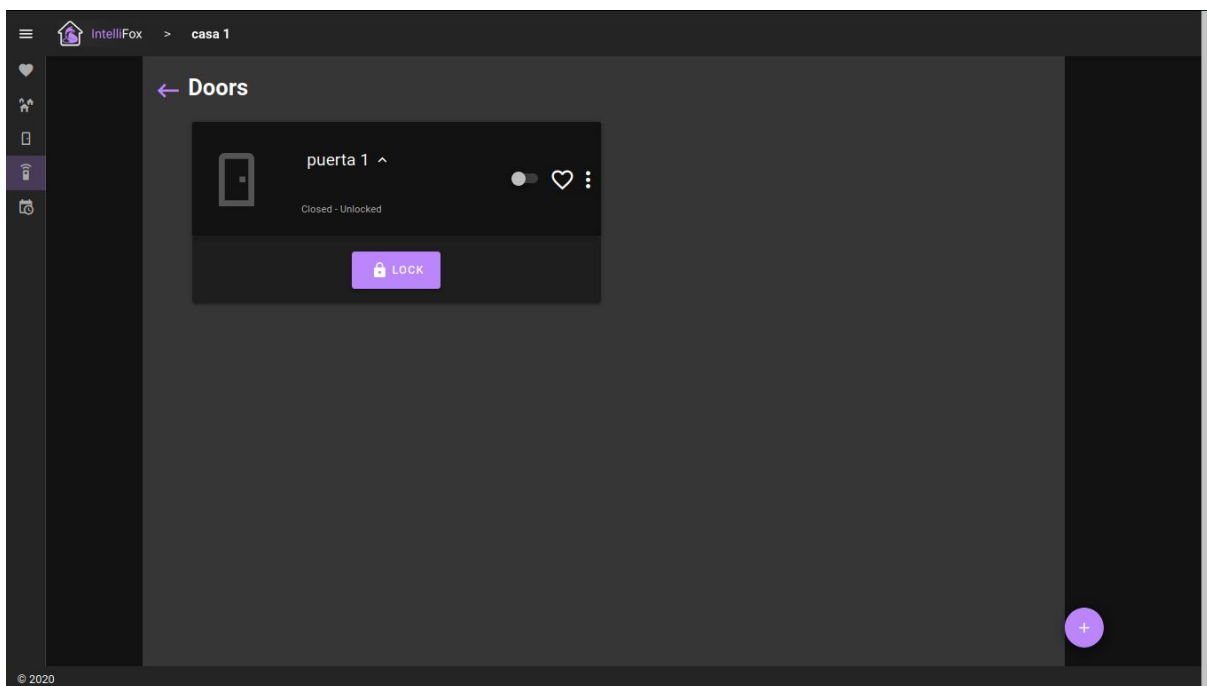


imagen 28 - Puertas

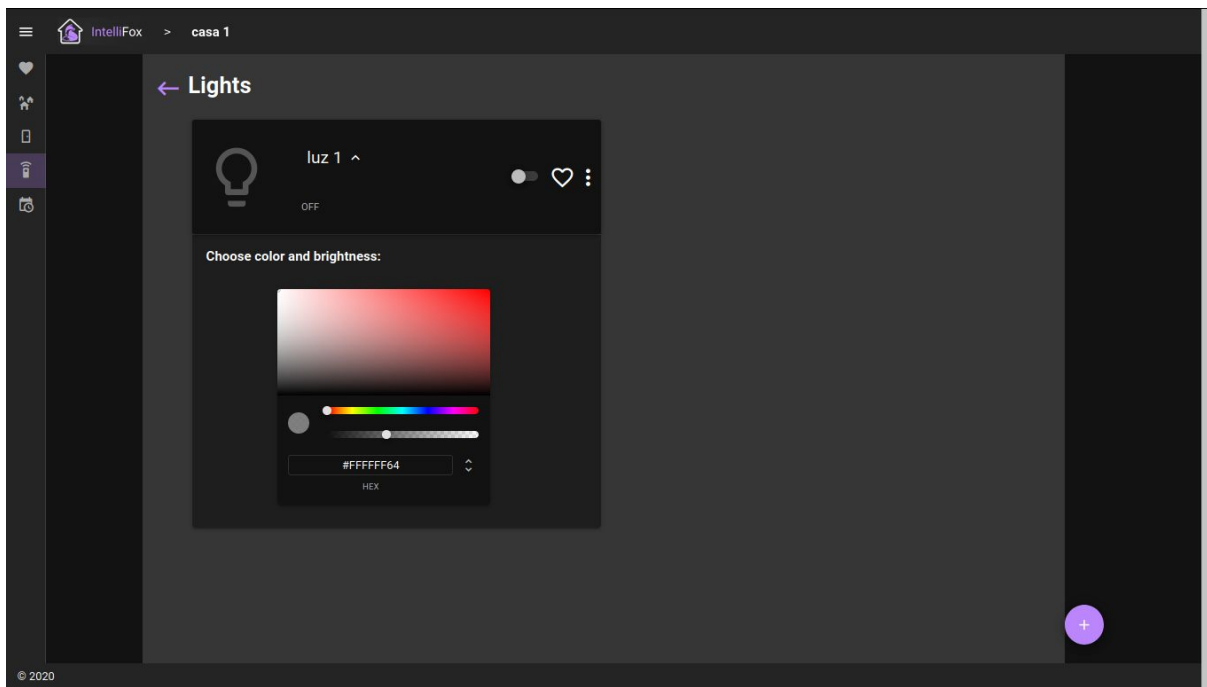


imagen 29 - Luces

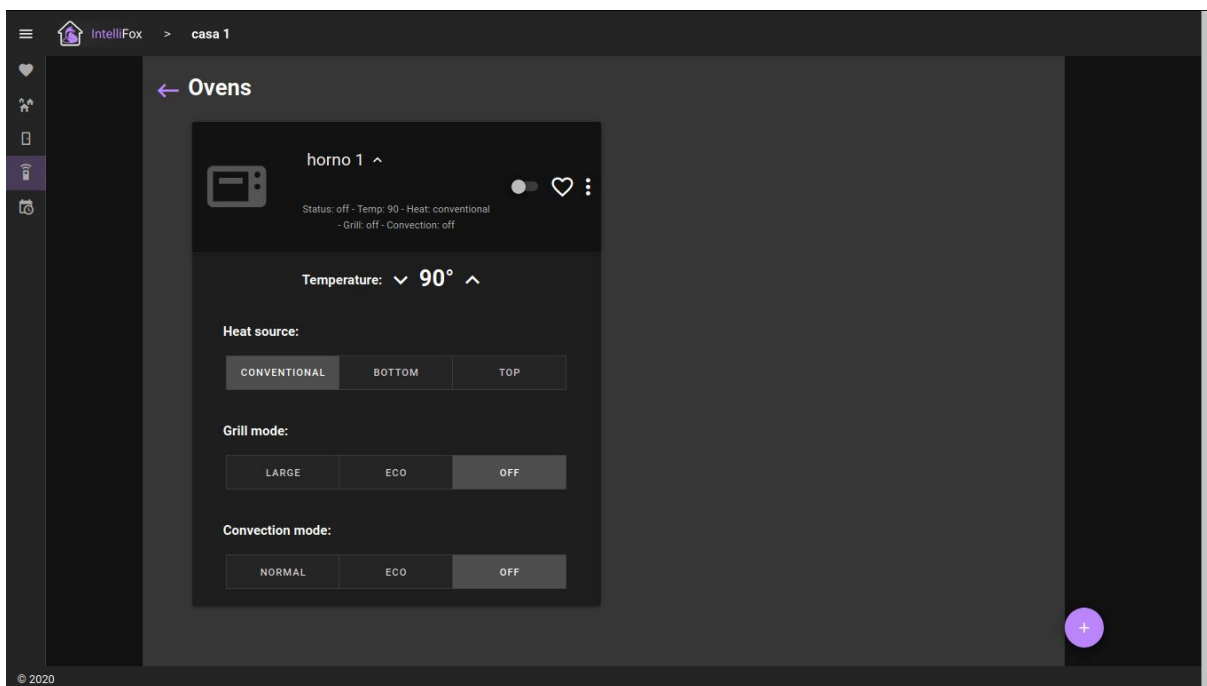


imagen 30 - Hornos

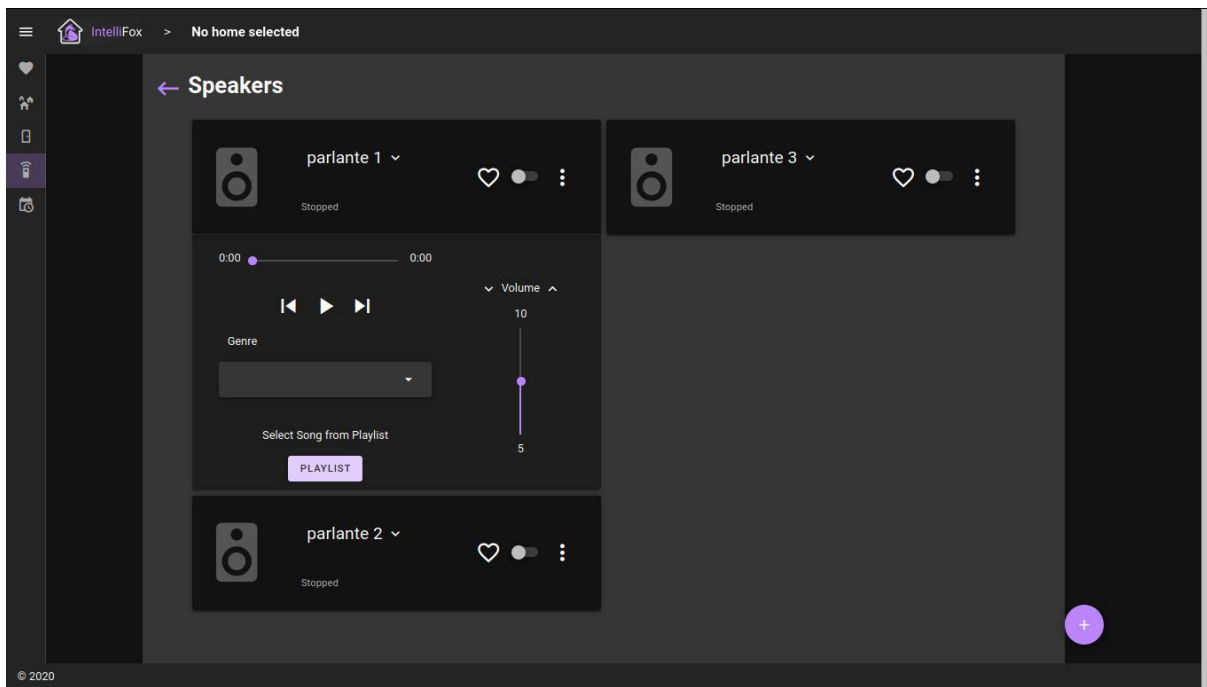


imagen 31 - Parlantes

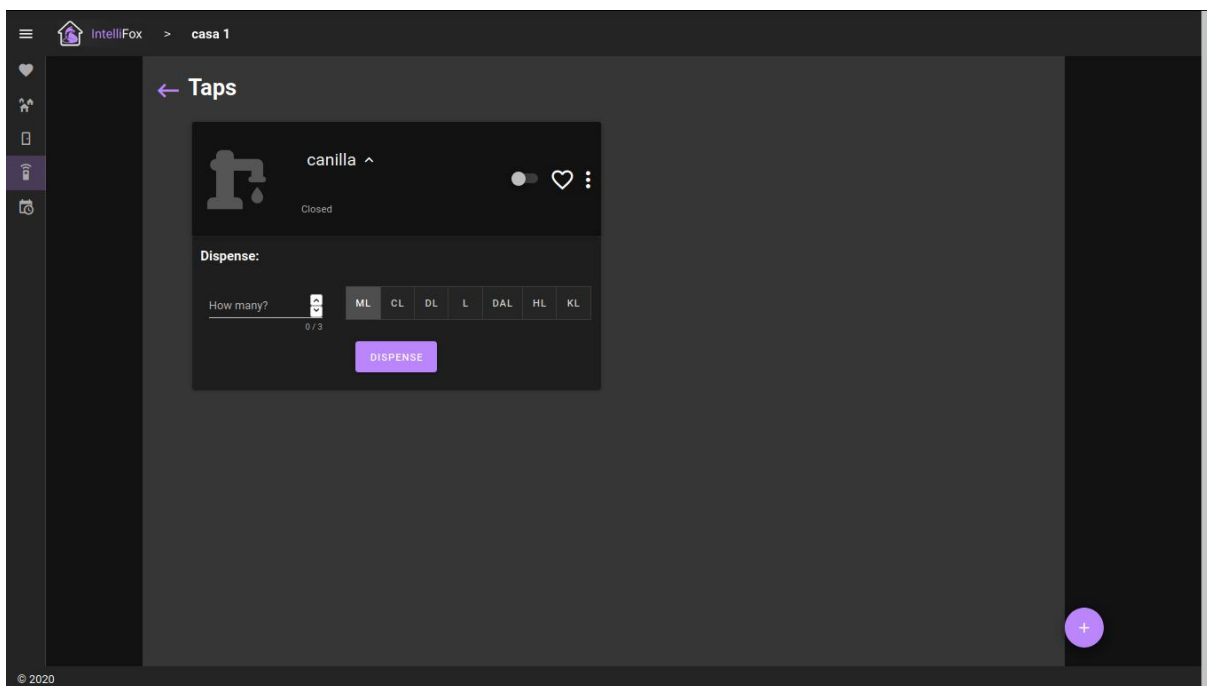


imagen 32 - Canillas

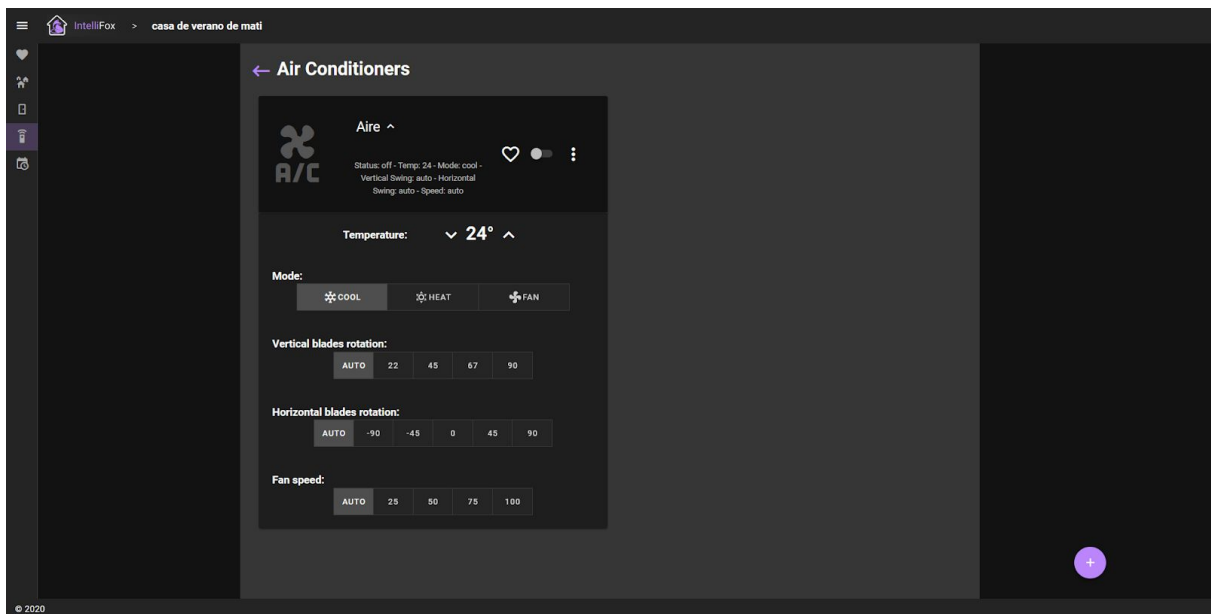


imagen 33 - Aires Acondicionados

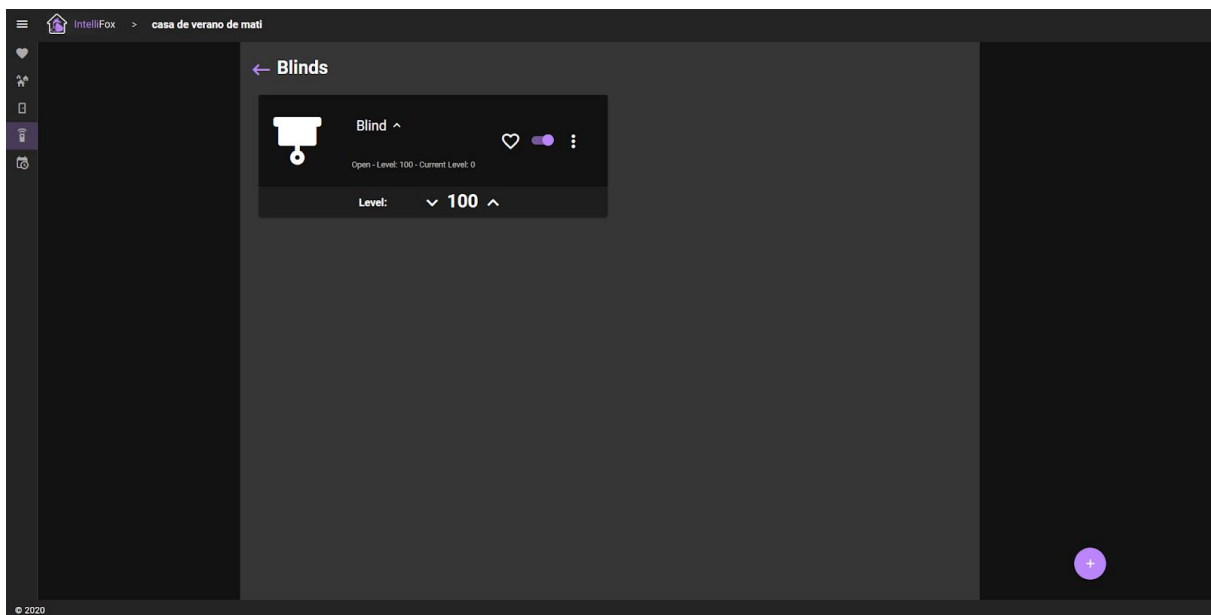


imagen 34 - Persianas

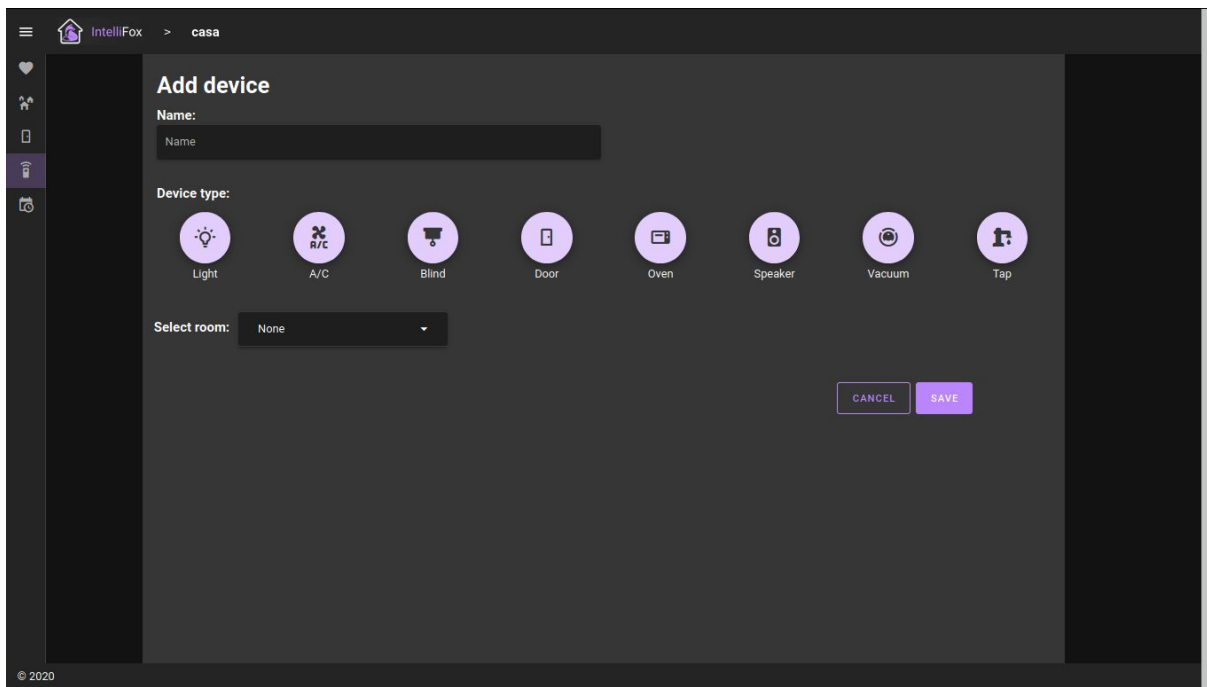


imagen 35 - Añadir dispositivos

4. Varios:

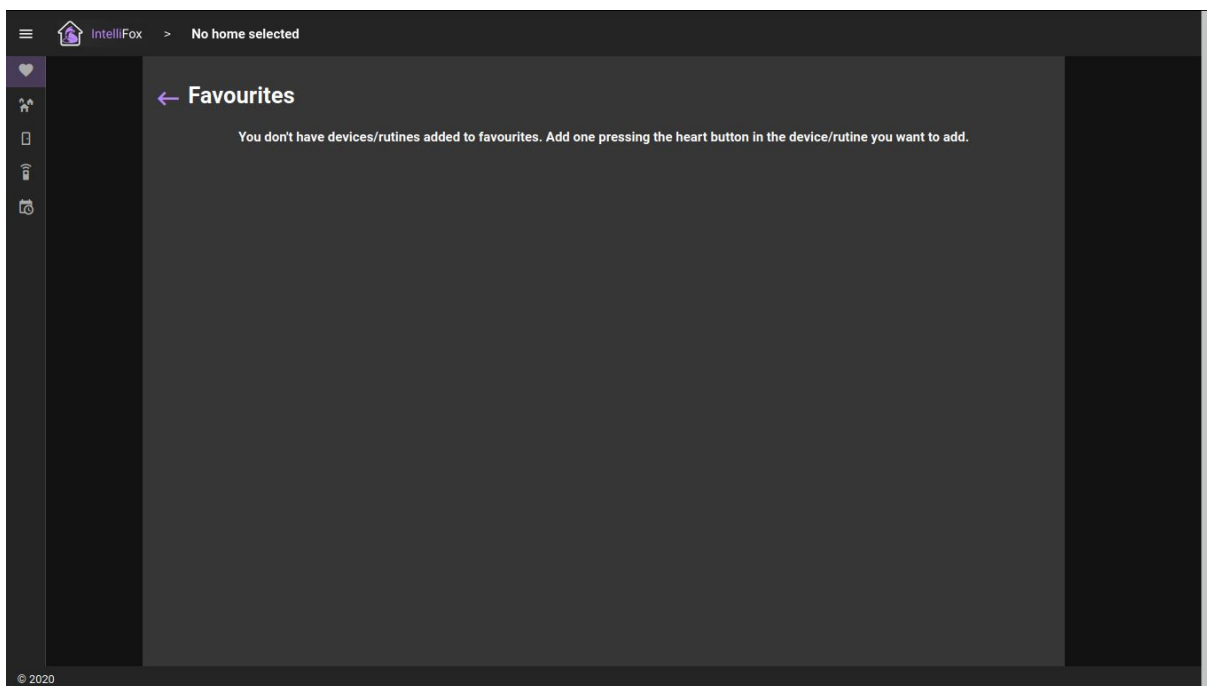


imagen 36 - Favoritos (vacío)

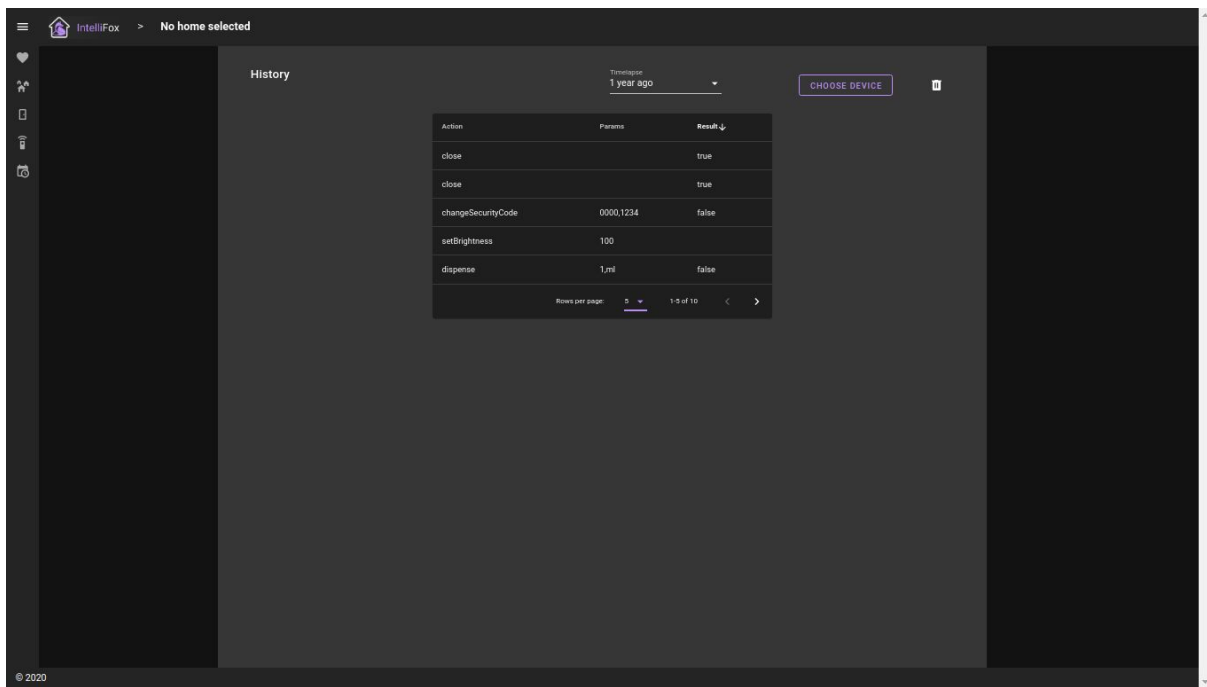


imagen 37 - Historial

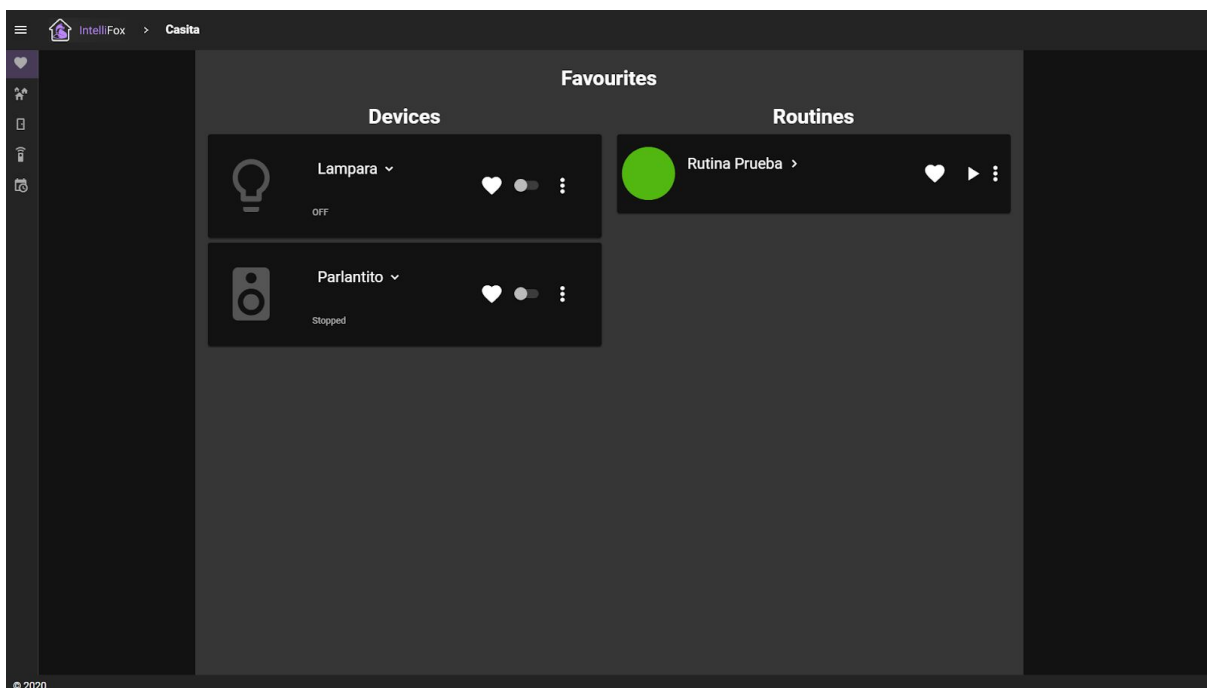


imagen 38- Favoritos

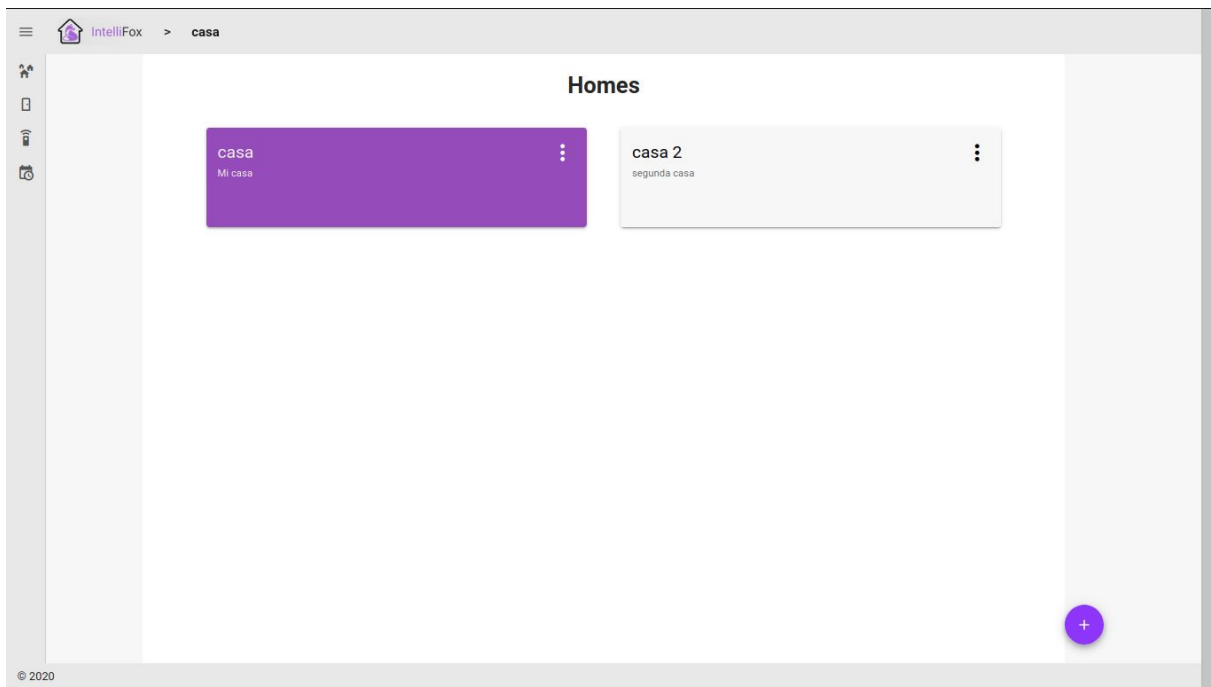


imagen 39- White mode

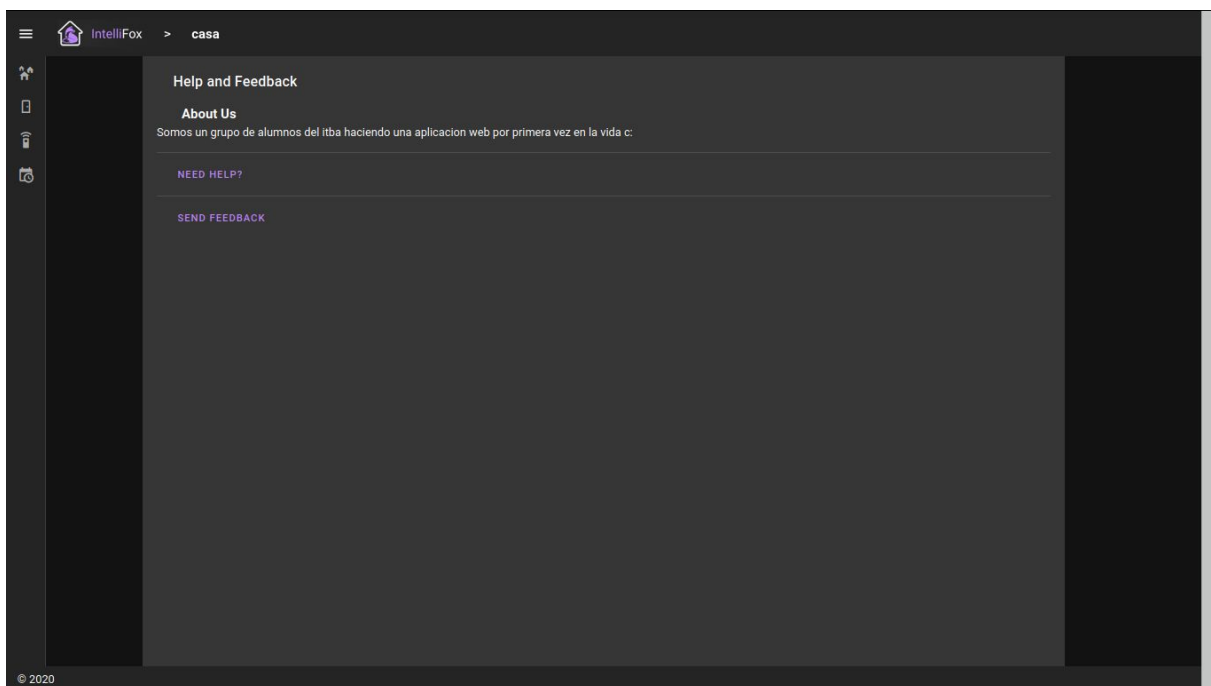


imagen 40- Help and Feedback

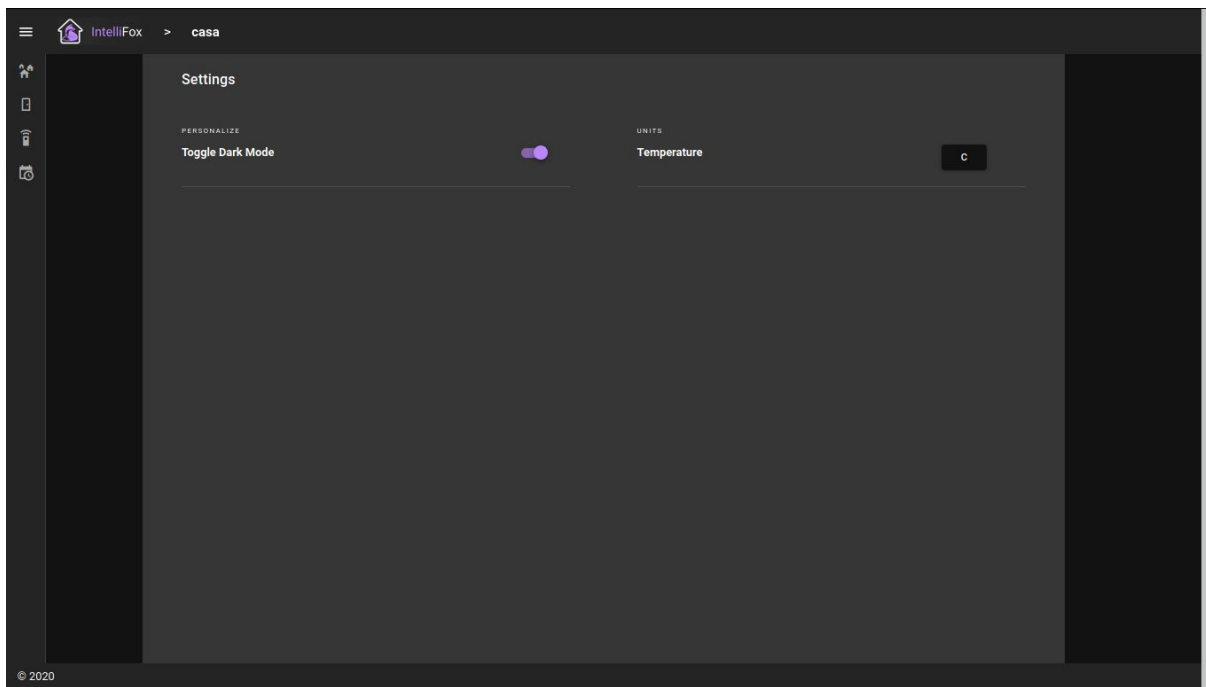


imagen 41- Settings

5. Rutinas:

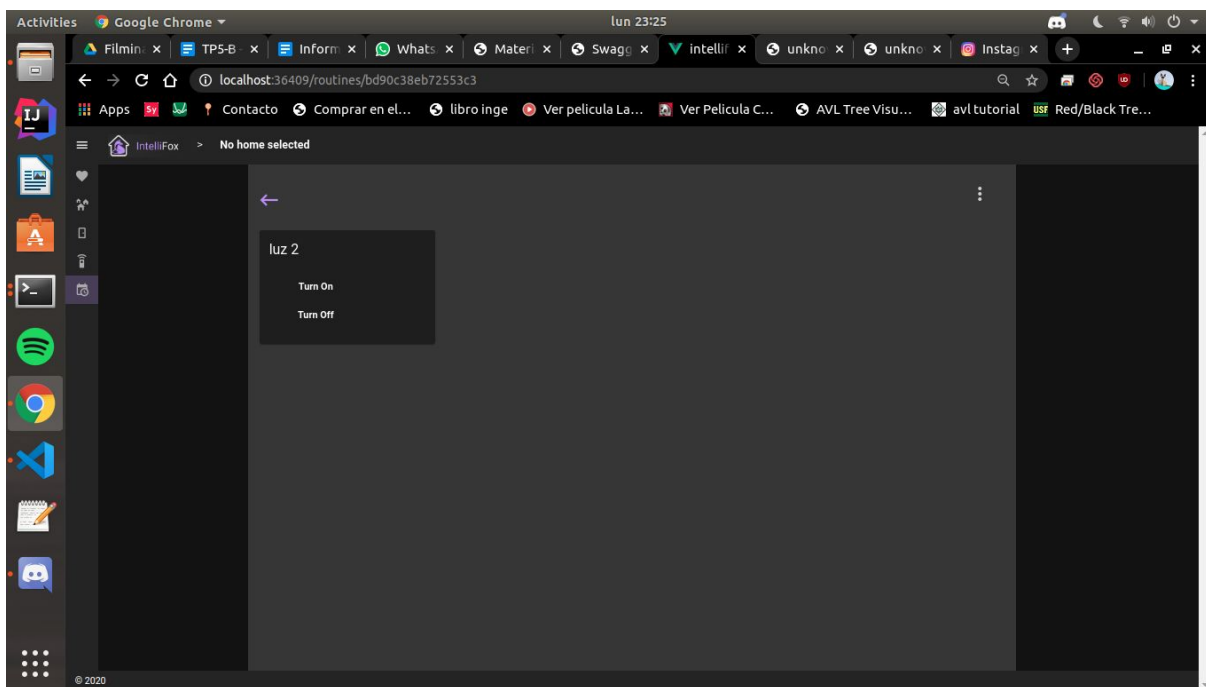


imagen 42- Dentro de una rutina

7) Problemas durante el desarrollo

Durante el desarrollo nos enfrentamos a varios inconvenientes al incorporar tecnologías nuevas para nosotros, como para entender cómo se estructura vue o cómo utilizar efectivamente los componentes de Vuetify. Por otro lado hubieron cosas que no pudimos implementar por cuestiones de tiempo. Cabe aclarar que muchas de estas funcionalidades no están implementadas de manera funcional pero si se muestran de forma visual.

- Cosas que no se llegaron a implementar:
 - Dispositivos
 - Heladera.
 - Alarma.
 - Funcionalidad
 - Consumo eléctrico.
 - PIN.
 - Error Log.
 - Edición de rutinas ya creadas.
 - Manejo de errores
 - No logramos implementar en su completitud un sistema eficaz para comunicarle al usuario cuando ocurre un problema como la pérdida de conexión o errores en el ingreso de datos.

8) Conclusión.

Utilizando las herramientas y los conocimientos brindados por la cátedra de forma incremental, recorrimos los procesos de creación de una página web pasando por HTML para la estructura, CSS para el estilo, JavaScript para la interacción y Ajax para los pedidos al API. Además, utilizamos los conocimientos teóricos vistos en la primera parte de la materia para hacer un sitio consistente y amigable al usuario, sin ruido visual ni información innecesaria para su uso.