



**Instituto Tecnológico de Buenos Aires**

**SEGUNDA ENTREGA: IMPLEMENTACIÓN DEL  
SITIO WEB**

*72.36 - Interacción Hombre-Computadora (HCI) - Grupo 10*

**Autores:**

**Bernasconi, Ian - 62867**

**Feferovich, Jeremías - 62701**

**Gutiérrez, Agustín - 62595**

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Requisitos Implementados</b>	<b>2</b>
2.1. Requisitos funcionales obligatorios . . . . .	2
2.2. Requisitos funcionales opcionales . . . . .	7
2.3. Requisitos no funcionales obligatorios . . . . .	9
<b>3. Capturas de pantalla de todas las páginas del sitio Web</b>	<b>10</b>
3.1. Pantalla de Inicio . . . . .	10
<b>4. Decisiones de usabilidad tomadas durante la etapa de implementación</b>	<b>13</b>
4.1. Diferencias presentadas en las páginas respecto de la versión final de los prototipos . . . . .	13
4.2. Implementación práctica de los temas vistos en la materia . . . . .	13
4.2.1. No hacemos pensar al usuario . . . . .	13
4.2.2. Navegación . . . . .	13
4.2.3. 8 reglas de oro . . . . .	14
4.2.4. Estándares de diseño . . . . .	14
4.2.5. Diálogos de confirmación . . . . .	15
4.2.6. Colores . . . . .	15
4.3. Implementación respecto a los modelos de persona . . . . .	15
4.4. Justificación haciendo referencia a las sugerencias y/o correcciones realizadas por los docentes. . . . .	16
<b>5. Archivos necesarios para el correcto funcionamiento del sitio Web</b>	<b>17</b>
<b>6. Instructivo de instalación</b>	<b>17</b>
<b>7. Conclusión</b>	<b>18</b>

## 1. Introducción

Este documento es la segunda de tres entregas que consisten en diseñar e implementar parcialmente un sitio Web y una aplicación móvil (Android) que permitan administrar y controlar en forma remota dispositivos inteligentes que se encuentren presentes en un hogar. Esta entrega se enfoca en la implementación del sitio Web.

Basándonos en las pautas que se establecieron en el primer informe, creamos el sitio web siguiendo las necesidades de los usuarios y los modelos de persona previamente definidos. De esta manera, el resultado obtenido se acerca bastante al prototipo final en cuanto a usabilidad y funcionalidad, aunque el diseño visual puede diferir. El diseño es amigable y fácil de usar, lo que coincide con el modelo mental que se espera de los usuarios representativos. Para lograr esto, se utilizaron varias tecnologías como HTML, CSS, Vuetify, JavaScript y VueJS.

## 2. Requisitos Implementados

### 2.1. Requisitos funcionales obligatorios

#### **Tipos de dispositivos soportados**

Los tipos de dispositivos soportados son los siguientes:

1. Aspiradora
2. Lámpara
3. Puerta
4. Horno
5. Aire acondicionado

Se pueden realizar todas las acciones relacionadas a estos dispositivos.

#### **Dispositivos**

En la sección 'Dispositivos', como se puede ver en la Figura 1, se puede crear un nuevo dispositivo con el botón de agregar.

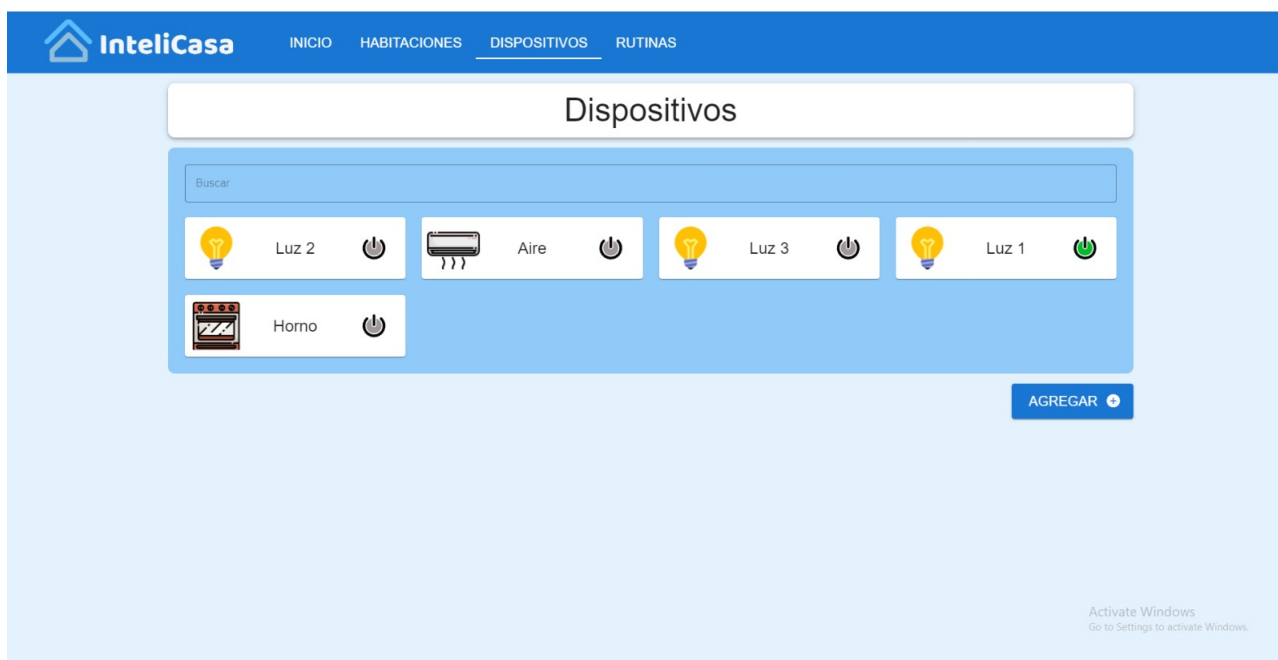


Figura 1: Pantalla de dispositivos

Una vez apretado el botón se despliega un *Dialog* que requiere de un nombre para el dispositivo y el tipo de dispositivo. Además, tiene la opción de agregarlo a una habitación existente o de crear una nueva habitación para este dispositivo. Ver Figura 2.

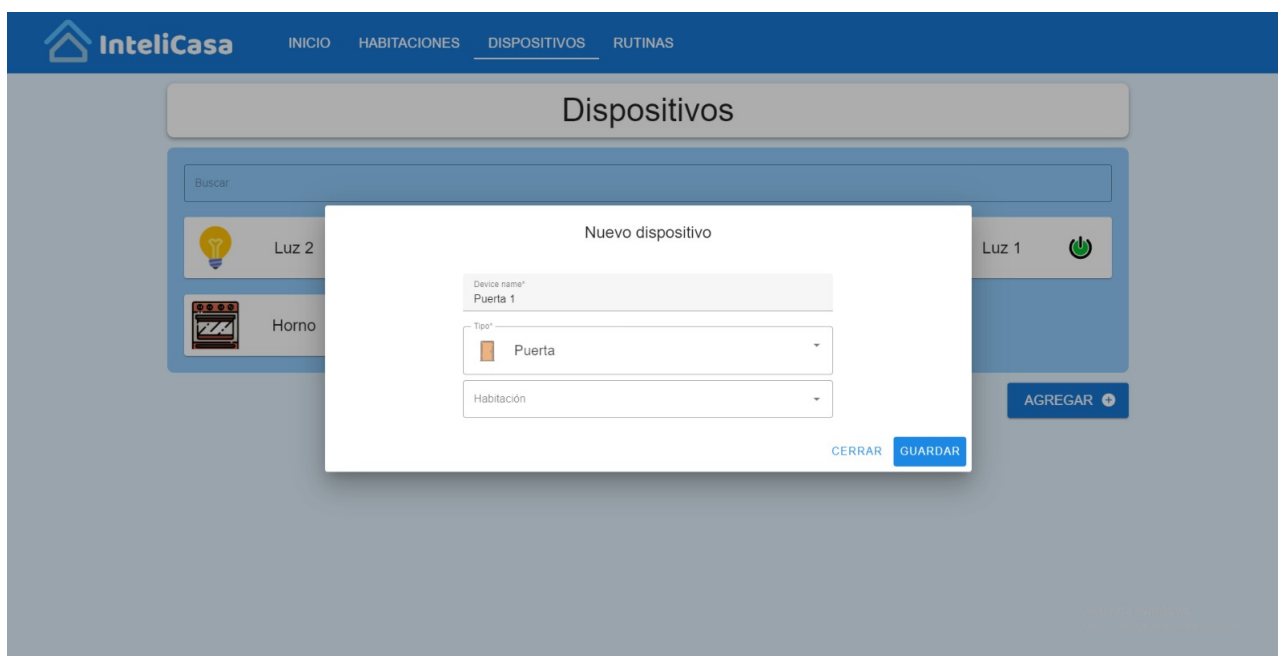


Figura 2: Pantalla de agregado de un dispositivo

Una vez creado, se puede ver en la misma pantalla el estado general de cada dispositivo. Haciendo click

en la *Card* de algún dispositivo se abre un *Dialog* que muestra el estado detallado del dispositivo y permite modificar el mismo, así como también cambiar su nombre, marcarlo como destacado o eliminarlo. Por ejemplo, para el caso de los dispositivos de tipo Lámpara, ver Figura 3 Se puede acceder a los dispositivos destacados rápidamente desde la página de ‘Inicio’.

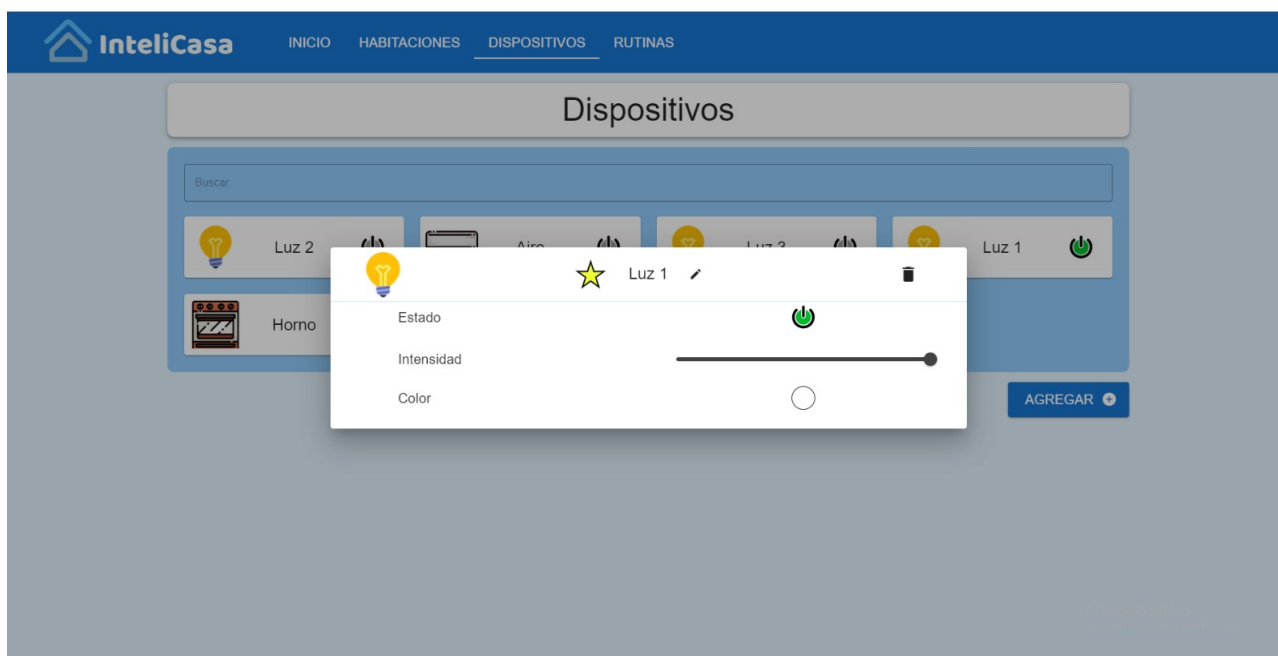


Figura 3: Pantalla de un dispositivo de categoría Lámpara

Antes de eliminar un dispositivo, así como en el resto de acciones no reversibles, se despliega un *Dialog* de confirmación, como el que se muestra en la Figura 4.

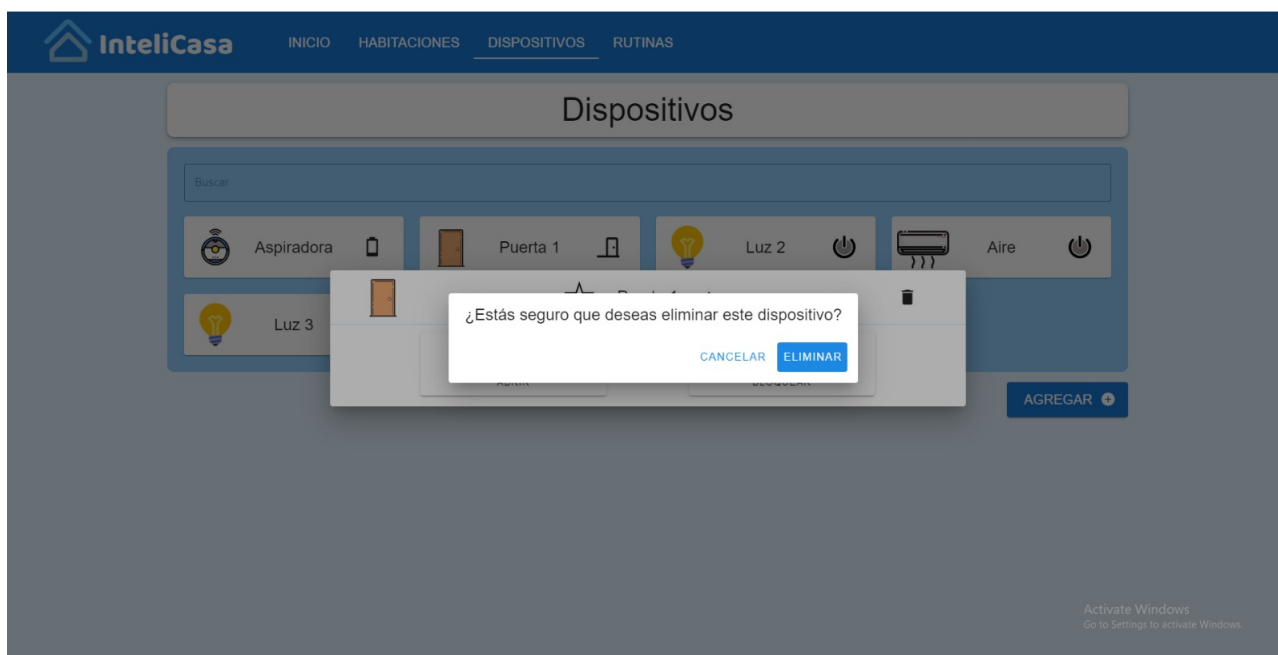


Figura 4: Pantalla de confirmación mostrada en el intento de borrar permanentemente un dispositivo

## Rutinas

En la sección 'Rutinas', se puede crear una nueva rutina con el botón de agregar, como se puede ver en la Figura 5.



Figura 5: Pantalla de Rutinas

Una vez apretado el botón se despliega un *Dialog* que requiere de un nombre para la rutina y los

dispositivos que se desean conectar a la misma. Los dispositivos aparecen en un menú desplegable y una vez seleccionado uno se puede especificar el estado en el que se quiere que ese dispositivo se encuentre luego de ejecutada dicha rutina. Ver Figura 6.

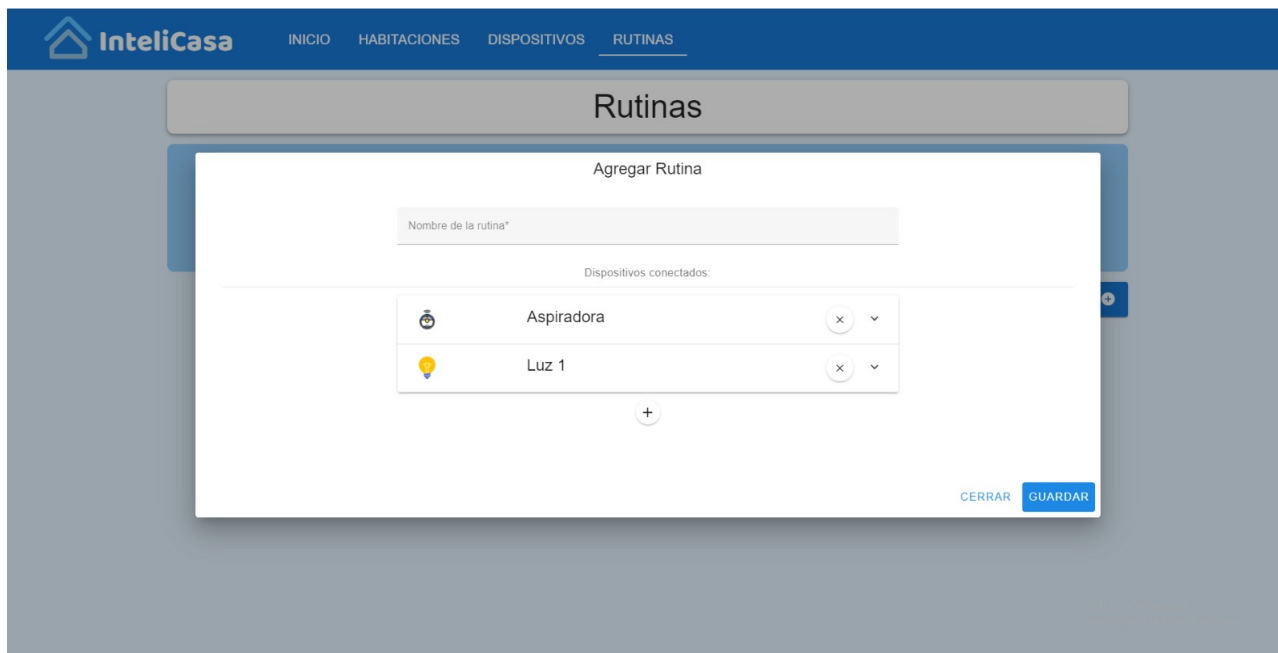


Figura 6: Pantalla de agregado de una Rutina

Una vez creada, se puede ver dicha rutina con las imágenes de los dispositivos conectados y un botón para ejecutarla. Haciendo click en la *Card* de alguna rutina se abre un *Dialog* que muestra los dispositivos vinculados a esa rutina, pudiendo modificar el estado objetivo de cada dispositivo, así como también eliminar o agregar dispositivos. Asimismo, se puede cambiar el nombre de la rutina, marcarla como destacada o eliminarla. Por ejemplo, para la rutina ‘Prender Luces’, ver Figura 7

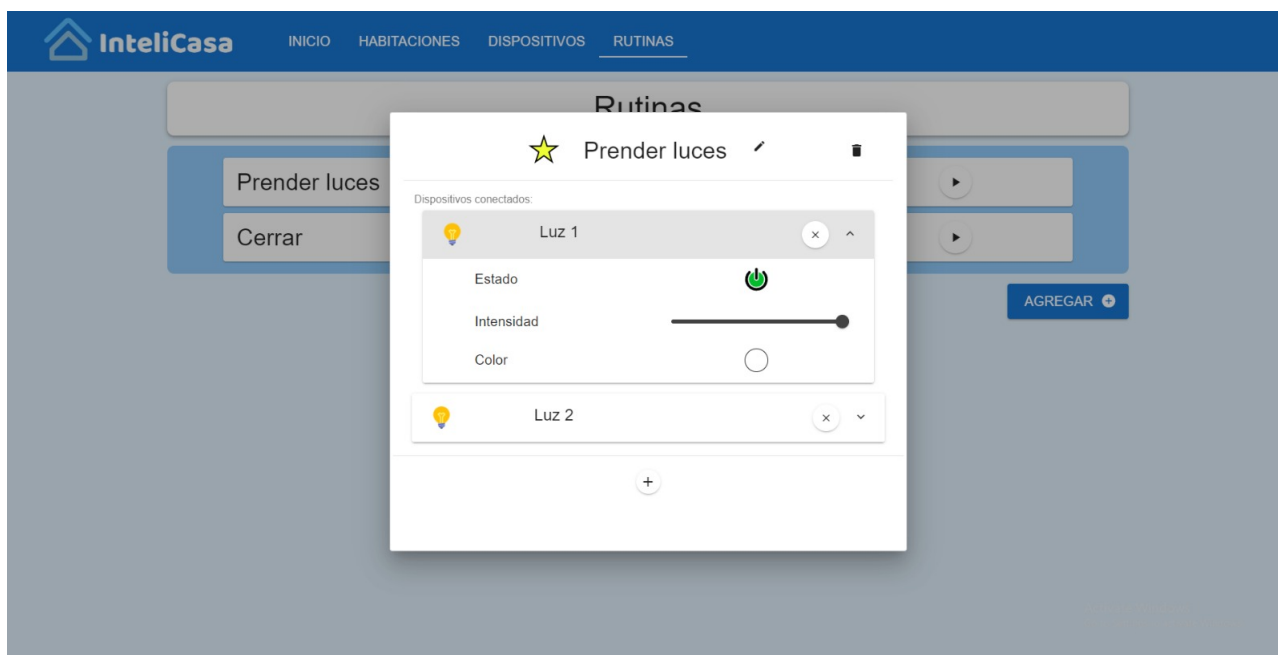


Figura 7: Pantalla de la rutina Prender Luces

Se puede acceder a las rutinas destacadas rápidamente desde la página de 'Inicio'.

## 2.2. Requisitos funcionales opcionales

### Habitaciones

En la sección 'Habitaciones', se puede crear una nueva habitación con el botón de agregar, como se puede ver en la Figura 8.





Figura 8: Pantalla de Habitaciones

Una vez apretado el botón se despliega un *Dialog* que requiere de un nombre para la habitación, los dispositivos que se desean conectar a la misma y el tipo de habitación. Ver Figura 9.

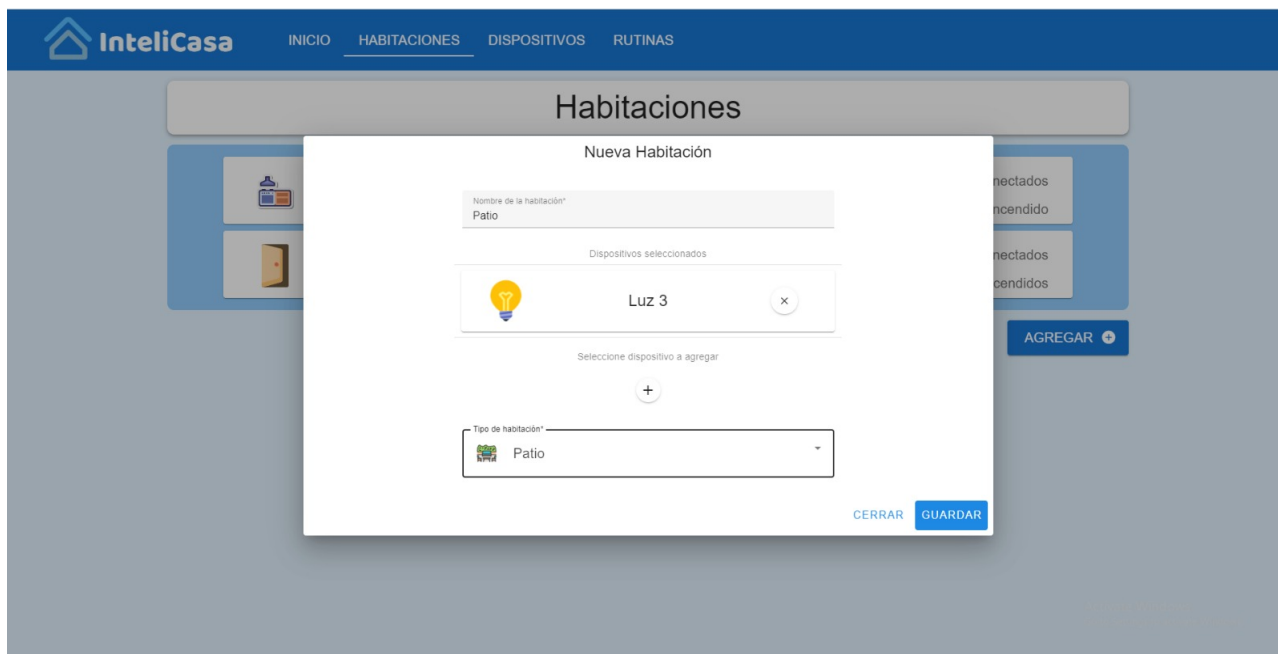


Figura 9: Pantalla de agregado de una habitación

Una vez creada, se muestra dicha habitación con la información de la cantidad de dispositivos conectados y encendidos a la misma. Haciendo click en la *Card* de alguna habitación se abre un *Dialog* que muestra

los dispositivos conectados a esa habitación, pudiendo modificar el estado de los mismos, así como también eliminar o agregar dispositivos. También se puede cambiar el nombre de la habitación o eliminarla. Por ejemplo, para la habitación ‘Cocina’, Ver Figura 10.

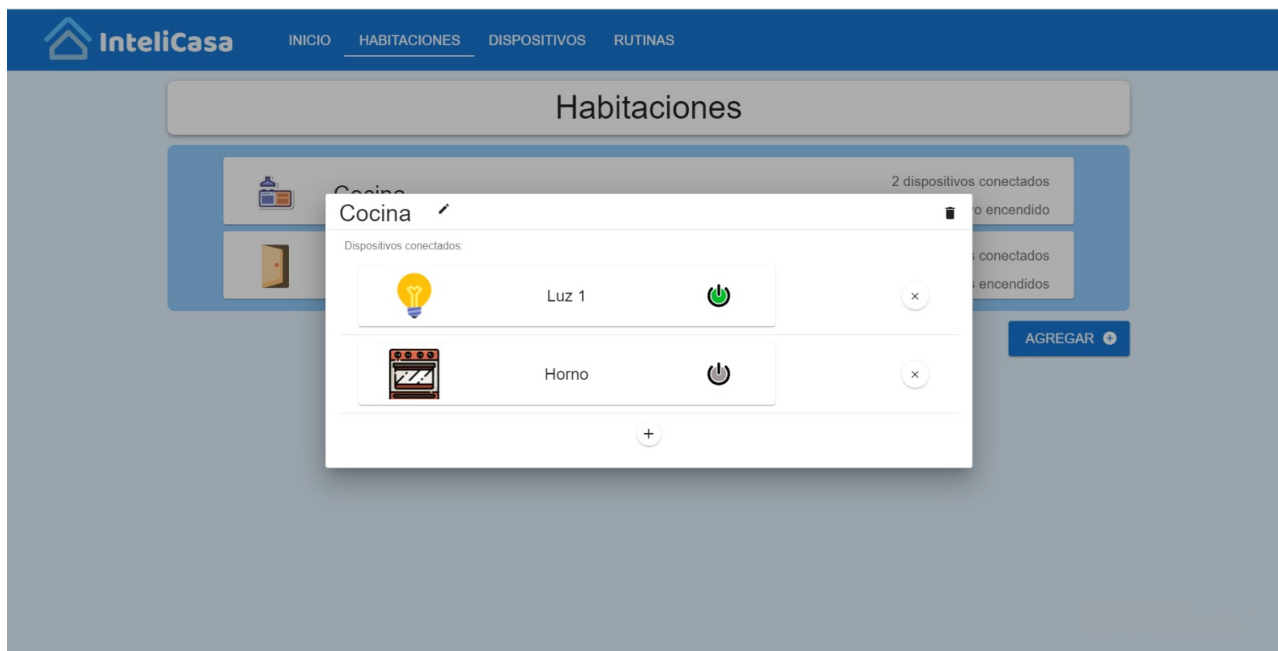


Figura 10: Pantalla de la habitación ‘Cocina’ y sus dispositivos

### 2.3. Requisitos no funcionales obligatorios

#### Uso de las tecnologías HTML, CSS y Javascript

Al utilizar Vue, estas tecnologías se ven utilizadas íntegramente, ya que Vue usa HTML para representar los componentes de la interfaz de usuario, CSS para aplicar estilos personalizados a los componentes y Javascript para proporcionar funcionalidad a los componentes de la aplicación.

#### Correcta visualización en los navegadores Microsoft Edge 40 (o superior), Mozilla Firefox 68 (o superior) y Google Chrome 81 (o superior)

La aplicación se visualiza correctamente en Microsoft Edge 113, Mozilla Firefox 113 y Google Chrome 113.

#### Correcta visualización en resoluciones de pantalla entre 1280px y 1920px

La página se visualiza como es esperado para pantallas dentro del rango provisto.

#### Respeto de las estrategias de separación entre la estructura y la presentación

Se intentó mantener separados la estructura y la presentación, definiendo la estructura dentro de los tags de `<template>` y la presentación dentro de `<styles>`, pero al utilizar la librería de componentes *Vuetify*, ciertos

elementos de presentación quedaron dentro de `<template>`, por ejemplo con el atributo `size` de los botones, y definiendo los márgenes dentro de `class`, como por ejemplo `ma-3`.

### Respeto de las estrategias de separación entre la estructura y el comportamiento

Se mantuvieron separados la estructura y el comportamiento, ubicándose la primera en `<template>` y el segundo en `<script>`

## 3. Capturas de pantalla de todas las páginas del sitio Web

### 3.1. Pantalla de Inicio



Figura 11: Pantalla de Inicio

En la pantalla principal de la web se pueden observar tanto las rutinas como los dispositivos destacados por el usuario para tener un fácil acceso a ellos. Las rutinas pueden ejecutarse mediante el botón de play, y se puede acceder a ellas haciendo click encima del card. En el caso de que no se hayan seleccionado rutinas destacadas o que no se haya agregado ninguna, se muestra un mensaje de información. Al hacer click sobre este, se llevará inmediatamente al usuario a la sección rutinas para que pueda agregar una rutina o seleccionar una como destacada. Dispositivos destacados también maneja esta última funcionalidad. Además, pueden prenderse o apagarse haciendo click sobre el botón de encendido o acceder a ellos con un click sobre el card.

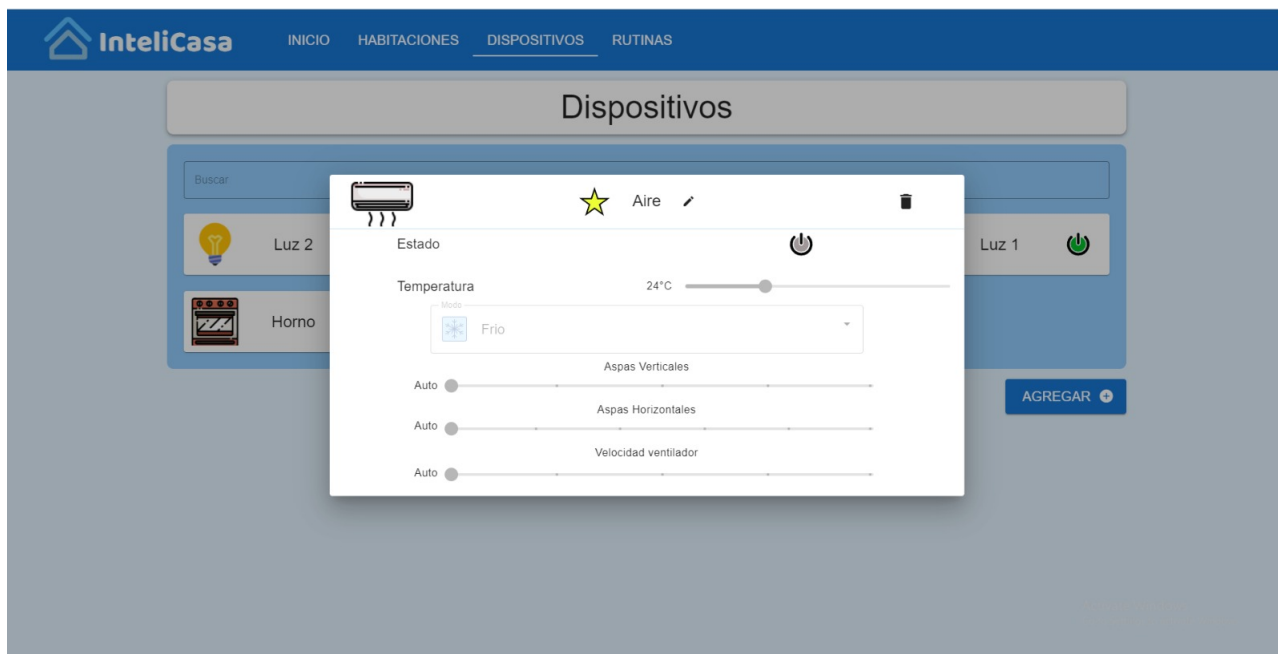


Figura 12: Pantalla de un dispositivo de categoría Aire

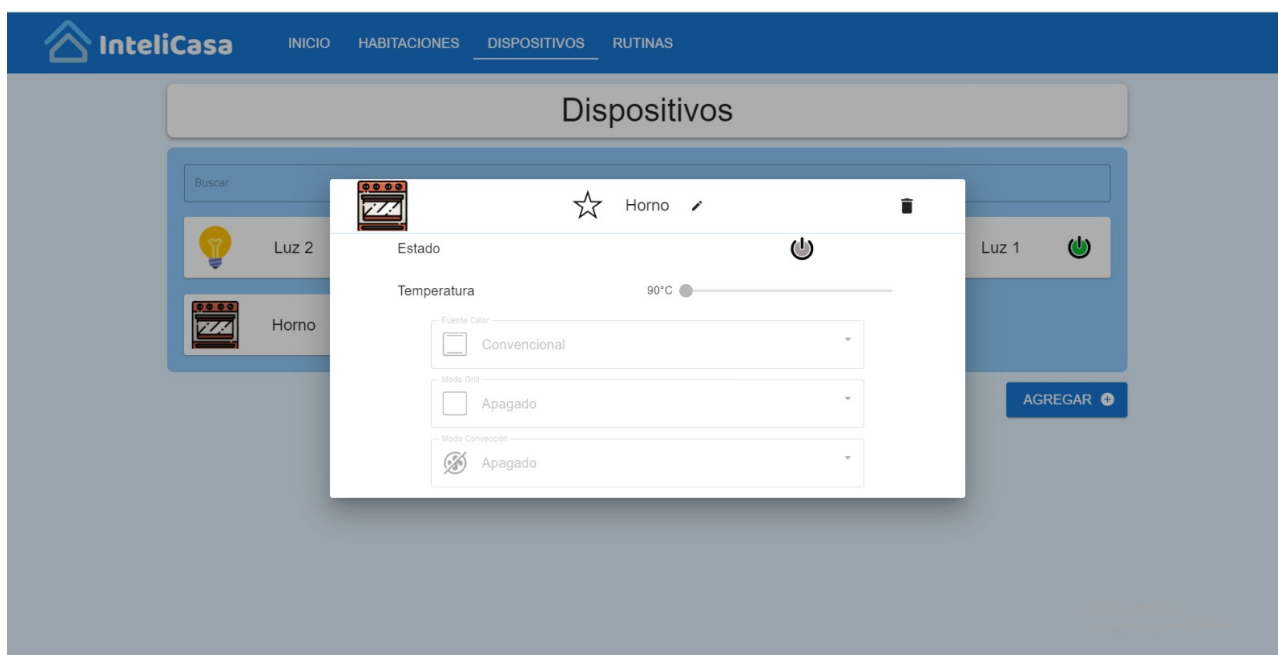


Figura 13: Pantalla de un dispositivo de categoría Horno

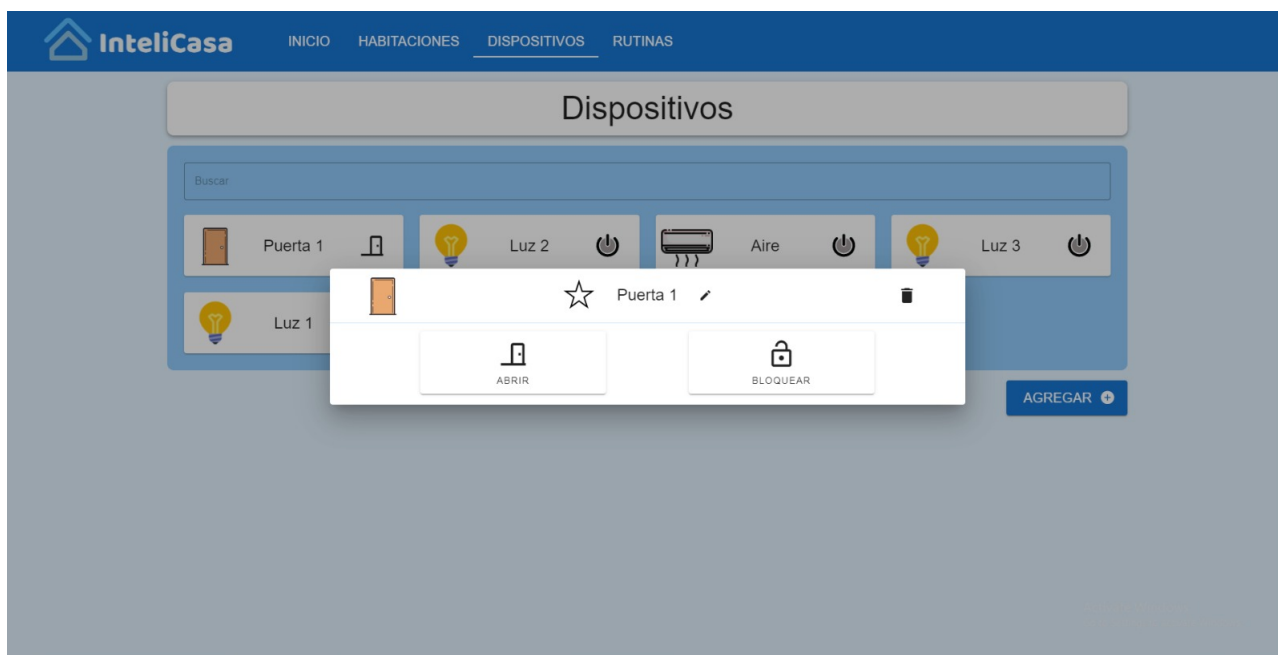


Figura 14: Pantalla de un dispositivo de categoría Puerta

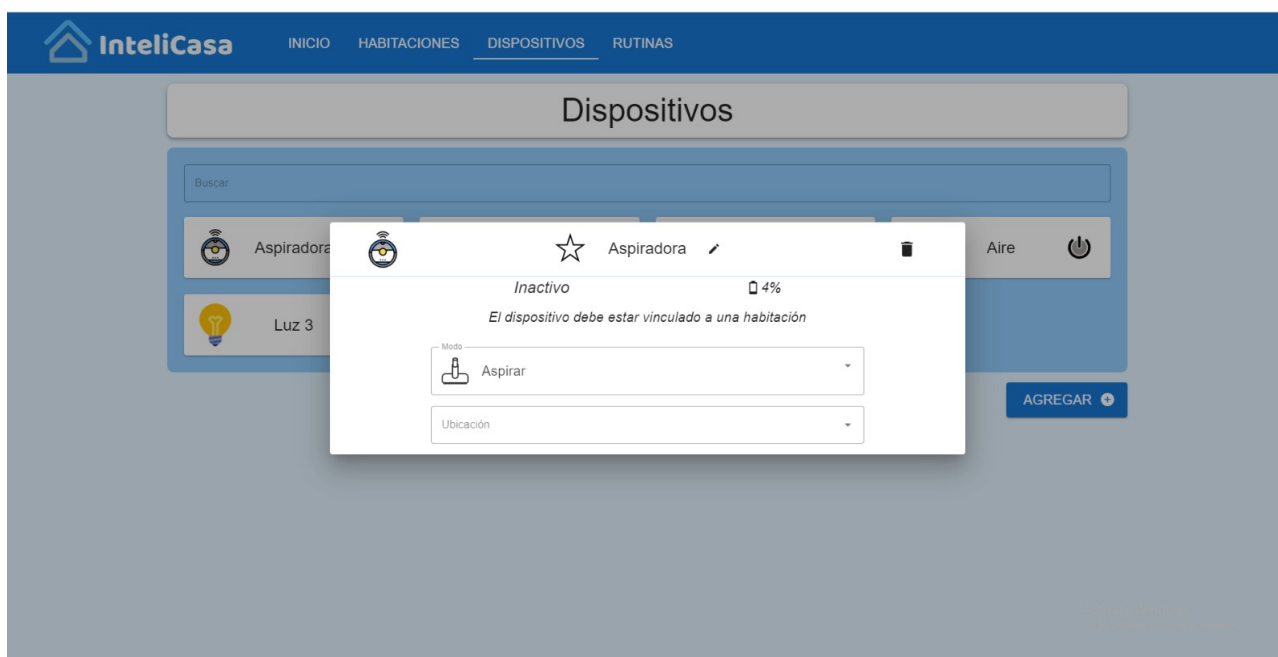


Figura 15: Pantalla de un dispositivo de categoría Aspiradora

## **4. Decisiones de usabilidad tomadas durante la etapa de implementación**

Se definió un máximo de 15 caracteres para los dispositivos, ya que consideramos que no son necesarios más para identificar a un dispositivo y esto nos permitió crear tarjetas más pequeñas, por lo que entran más en la pantalla.

Se decidió que si los dispositivos se encuentran apagados no sea posible modificar sus valores, ya que este cambio no tendría efecto.

### **4.1. Diferencias presentadas en las páginas respecto de la versión final de los prototipos**

En primer lugar, cabe destacar que existe una gran cantidad de cambios en el aspecto visual de la página en comparación con los prototipos finales. Esto se debe a que, al momento de diseñar los prototipos, no estábamos al tanto de las herramientas disponibles en Vuetify, que nos ofrecía diversas facilidades en términos de diseño. Un ejemplo es el componente de filtrado de dispositivos, que el equipo decidió remover. Por otro lado, la flecha de volver arriba en la izquierda en los diálogos de agregar un componente o información del mismo al hacer click. Esta funcionalidad fue reemplazada por el botón de cerrar cuando se intenta agregar un componente o hacer click fuera del diálogo cuando el usuario quiera interactuar con un componente ya creado.

### **4.2. Implementación práctica de los temas vistos en la materia**

#### **4.2.1. No hacemos pensar al usuario**

Cada botón en el que se puede hacer click presenta una sombra, elevación u otro color de fondo que lo hace fácilmente distinguible. Además, los campos de entrada de texto o de selección tienen un formato que corresponde al modelo mental comúnmente utilizado por los usuarios.

#### **4.2.2. Navegación**

La página posee una barra de navegación que está disponible en todas las pantallas principales de la web para que el usuario pueda manejarse por la aplicación de forma rápida y sencilla. Al momento de interactuar con acciones, como agregar un dispositivo, se mostrará el modal visto en las secciones anteriores. Si el usuario desea regresar a la pantalla principal de Dispositivos, debe hacer click fuera del diálogo y el componente no será creado. Asimismo, si el usuario desea interactuar con componentes ya existentes, se

mantiene la misma funcionalidad. En el caso de que el usuario intente acceder a una página inexistente, se decidió redirigir al usuario a la pantalla de Inicio.

#### 4.2.3. 8 reglas de oro

1. Consistencia: A través de toda la aplicación se mantiene consistencia en cuanto al formato, los botones, la estructura, etc.
2. Shortcuts: Los dispositivos y las rutinas a las que se quiera acceder más rápidamente se las puede destacar para así aparecer en la pantalla de inicio. Otro shortcut implementado es que cuando se crea un nuevo dispositivo y se le desea asignar a una habitación que no fue creada todavía, se le permite al usuario seleccionar la opción ‘Nueva Habitación’ y crearla en el momento.
3. Feedback: Todas las acciones que modifican el estado de un dispositivo tienen un feedback visual inmediato. Para cuando se corre una rutina, se implementó un *Snackbar* que muestra si la ejecución de la rutina fue exitosa o no.
4. Cierre de diálogos: Cuando se crea un dispositivo, una rutina o una habitación y se hace click en ‘Guardar’, el componente creado se muestra inmediatamente en pantalla, mostrándole al usuario que fue creado con éxito.
5. Manejo de errores: Todos los campos en los que el usuario puede ingresar información contienen restricciones que si son violadas muestran un mensaje de error.
6. Revertir fácilmente las acciones: Todo componente creado se puede borrar, y todo componente borrado puede ser creado de nuevo rápidamente.
7. Sentir el control: Al querer borrar un componente se muestra un diálogo de confirmación para confirmar que la acción que va a producirse es la deseada por el usuario.
8. Reducir lo que el usuario debe recordar: Las pantallas de la aplicación son simples y consistentes, lo que fomenta el reconocimiento de las mismas y la pronta familiarización con el funcionamiento. Además, el uso de dispositivos y rutinas favoritos disminuye los pasos necesarios para llegar a los mismos, así como también lo que se debe memorizar.

#### 4.2.4. Estándares de diseño

Es relevante destacar que durante todo el proceso de desarrollo utilizamos los estándares establecidos en Material Design y Vuetify con el objetivo de que nuestros usuarios se sientan cómodos utilizando interfaces

gráficas web que sean familiar para ellos. De esta forma, se utilizaron colores y tipografías que ya han sido utilizados y validados por la comunidad.

#### **4.2.5. Diálogos de confirmación**

En la primera sección del informe se discutió el tema de las acciones irreversibles, como por ejemplo eliminar una rutina. En estos casos, se suele mostrar un mensaje de confirmación al usuario para asegurarse de que realmente desea llevar a cabo la acción, ya que una vez realizada, no se puede deshacer y el usuario podría haber presionado el botón por error. En otras áreas del software, no se utiliza este mensaje de confirmación para que el usuario mantenga una sensación de control, dado que la presentación constante de mensajes de este tipo para cada acción podría generar una molestia o incomodidad en el usuario. Ya se vio en anteriores secciones como fue mostrado este mensaje en determinadas acciones.

#### **4.2.6. Colores**

Una parte crucial de la aplicación es la coherencia de la paleta de colores y su adecuación a los usuarios representativos. Nuestra paleta de colores es coherente con el diseño de Material Design y también con el producto al que está asociada. El azul se asocia comúnmente con la inteligencia, que en este caso hace referencia a un hogar inteligente. Es por eso que la aplicación utiliza este color como primario, mientras que los demás colores se han seleccionado basándonos en este para crear contrastes. Por otra parte, utilizamos iconos de colores saturados para que resalte a la categoría o tipo al que pertenece el componente. El resto de la web mantiene un diseño minimalista.

### **4.3. Implementación respecto a los modelos de persona**

En un principio, la aplicación está diseñada para que usuarios como María tengan fácil acceso a los dispositivos que más desee controlar en la pantalla de inicio. De esta manera, María puede controlar la intensidad de las luces rápidamente y obtener información en tiempo real de sus dispositivos destacados mediante un solo click. Usuarios como María tendrán un gran control sobre la aplicación y podrán experimentarla al máximo.

Además, la aplicación está diseñada para ser útil para usuarios como Miguel. Si el usuario no tiene ningún dispositivo creado, se muestra un mensaje de ayuda que lo dirige a la página donde puede agregar y configurar los dispositivos. Cada dispositivo tiene un ícono que representa la categoría a la que pertenece y otro que indica su estado actual. Una vez que el usuario tenga los dispositivos configurados y destacados, ya no es más necesario que el usuario navegue por las distintas pantallas, lo que otorga un funcionamiento sencillo y mecánico de la aplicación. Estas características están diseñadas para ayudar a Miguel a sentirse



más cómodo y seguro al utilizar la aplicación web.

En cuanto a usuarios con deseos similares a los de Juan, la aplicación está estructurada de manera simple a través de una barra de navegación, y la identificación de tipos de dispositivos y sus estados con iconos, los que permiten proporcionar buen feedback al usuario. Esto hace que la aplicación sea fácil de usar y no requiera conocimientos técnicos para sus características principales. Juan podrá mantener su casa segura al controlar el dispositivo de la puerta, aunque se sentirá decepcionado al descubrir que la aplicación no admite dispositivos de alarma o sensores de movimiento.

#### **4.4. Justificación haciendo referencia a las sugerencias y/o correcciones realizadas por los docentes.**

- *Tener en consideración que existe una falta de consistencia respecto del uso de iconos junto a las leyendas de las secciones. Por otro lado, algunos usuarios podrían verse confundidos por la presencia del botón de reproducción junto a la leyenda “Rutinas”.*

En cuanto al icono de reproducción en la página de inicio junto a la leyenda Rutinas, decidimos removerlo para evitar confusiones, dejando como leyenda Rutinas Destacadas.

- *Tener en consideración la posibilidad de estandarizar la forma en la que se muestran los títulos y/o secciones a lo largo de las diferentes páginas.*

Se decidió estandarizar el tamaño y la ubicación de los títulos y secciones en todas las páginas de la aplicación con el objetivo de mantener una consistencia constante en todo el sitio y asegurarse de que estén bien alineados entre sí.

- *Tener en consideración la posibilidad de representar las tarjetas de rutinas de una forma similar a las tarjetas de dispositivos*

Se decidió cambiar color de las tarjetas de rutinas en el inicio para que sean más similares a las tarjetas de dispositivos, dejando una forma diferente para marcar que son elementos diferentes.

- *Tener en consideración que existe una falta de consistencia respecto al tamaño, forma y color usado para representar los botones.*

Se implementaron botones generales para cada funcionalidad (agregado de componentes, guardado, cerrar que permiten mantener una consistencia a lo largo de la página.

- *Tener en consideración la posibilidad de posicionar el botón que permite eliminar un dispositivo en la parte superior de la pantalla.*

Esto fue implementado en la parte superior derecha de la pantalla en todos los componentes que tienen la funcionalidad de ser borrados.

- *Tener en consideración que las rutinas son de ejecución inmediata y, por lo tanto, no tienen un estado. Por otro lado, la ejecución de una rutina en forma planificada o ante un cambio en el estado*

Se decidió cambiar el botón de ejecución de una rutina en el prototipo por uno de play para evitar confusiones y desvincular la opción de estado de una rutina, ya que las rutinas son de ejecución inmediata y no tienen un estado, aunque pueden ser ejecutadas de forma planificada o en respuesta a un cambio de estado.

- *Mal por utilizar dos términos para hacer referencia al mismo concepto (“crear” y “agregar”).*  
El grupo optó por la opción de “agregar” un componente.

## 5. Archivos necesarios para el correcto funcionamiento del sitio Web

Para el funcionamiento del sitio web, son necesarios los archivos de la carpeta `/src`. En la carpeta `/src/api` se encuentran los archivos necesarios para comunicarse con la API, en `/src/assets` se encuentran los iconos e imágenes utilizados a través de la aplicación, en `/src/stores` se encuentran los archivos necesarios para vincular *pinia* con la API, y en `/src/views` las vistas principales de cada página del sitio. Por otro lado, en `/src/components` se encuentra el código de los componentes utilizados a través de la aplicación.

## 6. Instructivo de instalación

A continuación se detallan las instrucciones para correr la aplicación web en un entorno local.

- Correr la API:
  1. Descargar el archivo de la API y descomprimirlo
  2. Instalar las dependencias necesarias con el comando `npm install`
  3. Correr la API abriendo una terminal en la carpeta donde se encuentra y ejecutando el comando `npm start`
- Correr el sitio web
  1. Descargar el archivo y descomprimirlo
  2. Instalar las dependencias necesarias con el comando `npm install`

3. Instalar *serve* con el comando *npm install -g serve*
4. Construir la página con *npm run build*. Se genera en la carpeta *dist*
5. Correr la página con el comando *serve -s dist*
6. Correr la API abriendo una terminal en la carpeta donde se encuentra y ejecutando el comando *npm start*

■ Notas

- La URL de la API se puede modificar en el archivo */src/api/api.js*

## 7. Conclusión

Durante el desarrollo de este trabajo, comprendimos la agilidad a la hora de la implementación que nos proporcionó tener un prototipo definido de antemano, ya que nos evitó tener que discutir cuestiones del diseño general para así poder enfocarnos en el desarrollo. La falta de implementación de la API desde el principio ralentizó el proceso de desarrollo y originó grandes cambios en nuestro código, su organización en módulos y la interacción con ella.