

Trabajo Práctico HCI - Grupo 6

Implementación de la aplicación móvil

62466 - Fernández Dinardo, Juan Ignacio

jfernandezdinardo@itba.edu.ar

61258 - Panighini, Franco

fpanighini@itba.edu.ar

60044 - Thomas, Park

tpark@itba.edu.ar

61007 - Rivas Betancourt, Santiago

srivas@itba.edu.ar

1. Requisitos funcionales implementados	3
1.1 Navegación	4
1.2 Dispositivos, sus estados y configuraciones	6
1.2.1 Luz	7
1.2.2 Horno	8
1.2.3 Aspiradora	9
1.2.4 Aire Acondicionado	10
1.2.5 Aspersor	11
1.3 Consultar las rutinas	13
1.4 Notificaciones	15
2. Requisitos no funcionales implementados	16
2.1 Idiomas	16
2.2 Personalización	16
2.3 Factor de Forma	17
2.4 Orientación del dispositivo	19
2.5 Funcionar en dispositivos Pie 9.0 - API Level 28 (o superior)	20
3. Decisiones de usabilidad tomadas durante la etapa de implementación	20
3.1 Cambios en los prototipos	20
3.2 Colores	26
3.3 Globalización	27
3.4 Feedback	27
4. Limitaciones	29
5. Archivos necesarios para construir el instalador de la aplicación	30
6. Instructivo de instalación de la APK	30
7. Conclusión	31
8. Anexo	32
8.1 Vistas en modo oscuro	32

Introducción

El presente informe describe la última entrega del diseño e implementación de una aplicación de control de dispositivos de smart home, tanto a través de un sitio web como de una aplicación móvil. En esta entrega, nos enfocamos en el desarrollo de la aplicación móvil, utilizando tecnologías avanzadas como Jetpack Compose para ofrecer una experiencia de usuario mejorada.

Nuestro objetivo principal es crear una aplicación amigable y predictiva, que se ajuste al modelo mental de los usuarios representativos. Para lograr esto, hemos utilizado el lenguaje Kotlin y aprovechando las capacidades de Jetpack Compose, una biblioteca moderna y poderosa que simplifica el desarrollo de interfaces de usuario dinámicas y receptivas.

Mediante Jetpack Compose, hemos logrado diseñar una interfaz intuitiva y atractiva, donde los usuarios pueden controlar sus dispositivos de smart home de manera sencilla y eficiente. Además, Jetpack Compose nos ha permitido aprovechar las ventajas del enfoque declarativo, facilitando la creación de componentes reutilizables y la gestión del estado de la aplicación.

En conjunto con las herramientas de Android Studio, como Gradle y el emulador de Android, hemos logrado una implementación eficiente y efectiva de la aplicación. Además, hemos utilizado el backend proporcionado por la cátedra, accediendo a sus funcionalidades a través de una API, lo que nos ha permitido enfocarnos en la capa de presentación y la experiencia del usuario.

En resumen, esta entrega presenta una aplicación de control de dispositivos de smart home desarrollada utilizando Kotlin, Jetpack Compose y herramientas de Android Studio, con el objetivo de brindar una experiencia de usuario intuitiva y satisfactoria.

1. Requisitos funcionales implementados

En este apartado se explicarán las decisiones tomadas en cuanto al desarrollo de los requisitos funcionales de la aplicación móvil y el funcionamiento de estas. Más adelante, en la sección de usabilidad, se exhibirán las diferencias entre las distintas vistas implementadas y sus respectivos prototipos.

1.1 Navegación

En primer lugar, al igual que el sitio web, la aplicación ha sido desarrollada para tener un acceso rápido y directo a todas las vistas principales (“places”, “devices”, “routines”, “favourites”) para que el usuario no tuviese que hacer demasiado esfuerzo recordando cómo encontrar cada elemento. Es por ello que la barra inferior horizontal, propiamente dicha barra de navegación, se encuentra el acceso a estas vistas.



Imagen 1.1.1 Navigation Bar

La vista principal o de inicio es la de “Devices” para que el usuario pueda tener rápidamente una vista general de todos los dispositivos sincronizados. Dentro de la misma, yace un “carrousel” en la parte superior de la pantalla que le permite al usuario filtrar por el tipo de dispositivo. Este agregado cumple la función del menú lateral izquierdo que posee el sitio web. Análogamente, en la vista o sección “Places” el usuario puede filtrar a los dispositivos según el ambiente en el que estos se encuentran, mientras que en “Routines” simplemente se despliega una vista con todas las rutinas creadas y la posibilidad de ejecutarlas. Finalmente, en “Favorites” se encontrarán aquellos dispositivos que el usuario considere más importantes o de uso frecuente.

Si retrocedemos a los primeros prototipos, encontramos que esta misma barra estaba presente:

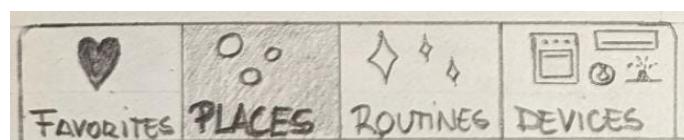


Imagen 1.1.2 Primer prototipo de navigation bar.

Como se puede ver, únicamente se cambió el orden de las secciones y se seleccionaron logos más representativos.

Volviendo a nuestra app actual, a continuación se muestran ejemplos de las diferentes secciones de la barra de navegación:

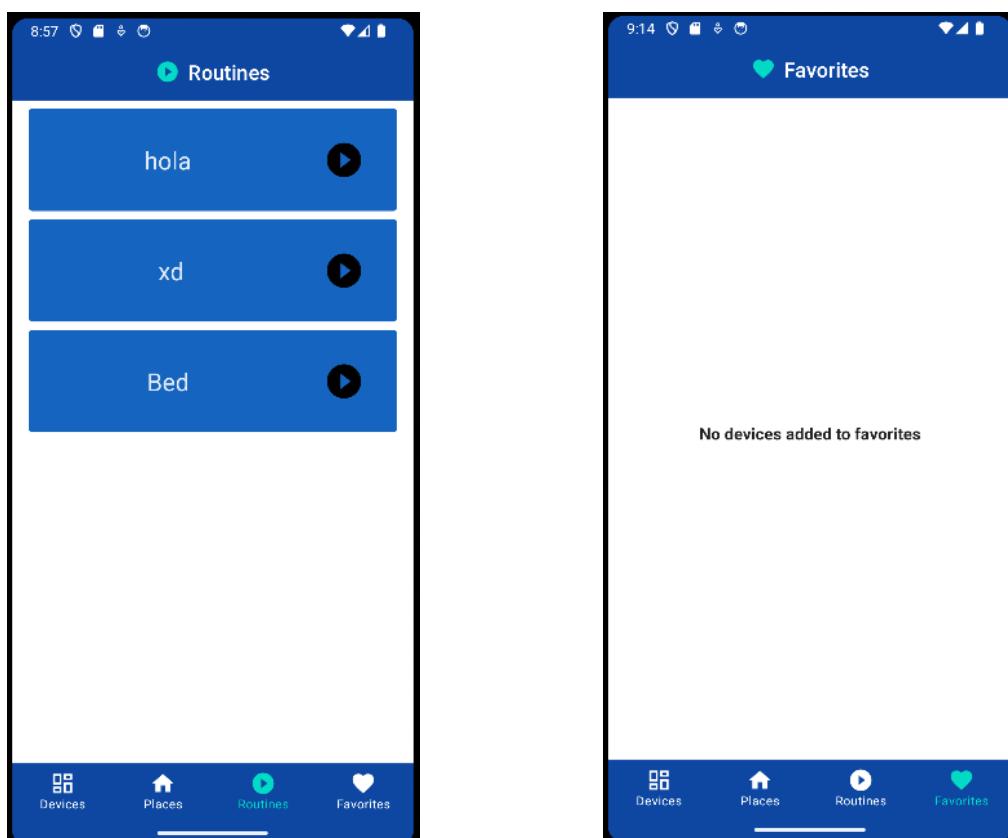
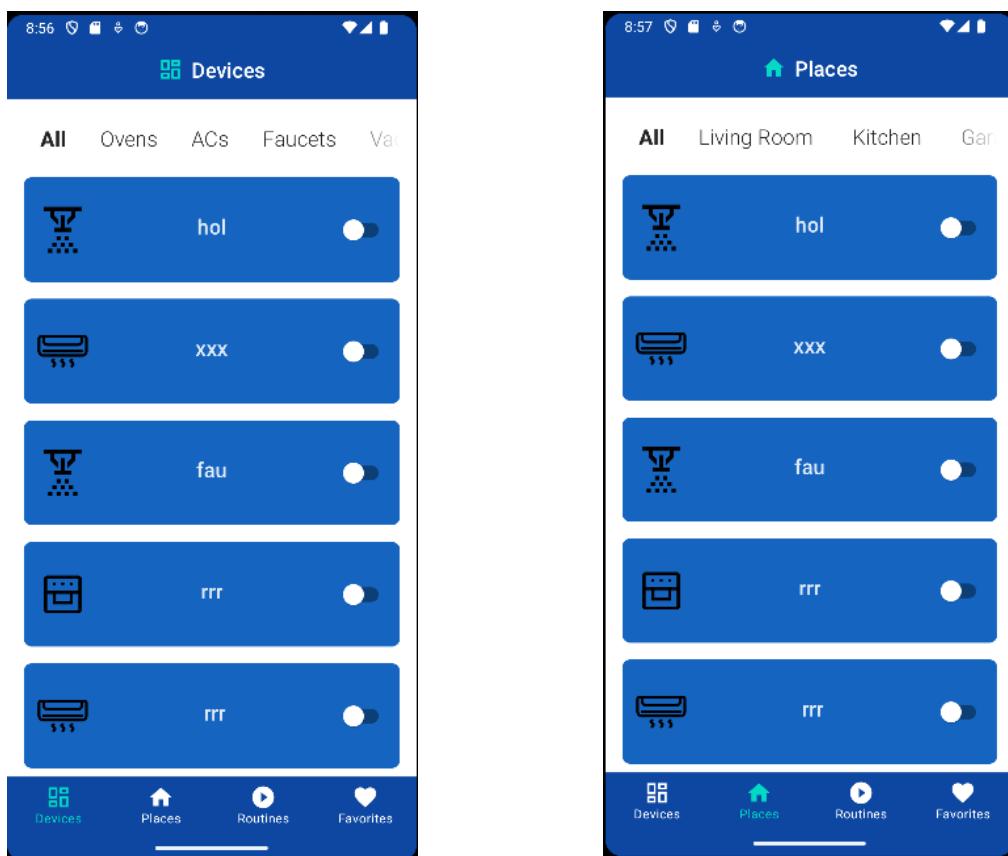


Imagen 1.1.3 Vistas de las diferentes secciones: Devices, Places, Routines, Favourites

1.2 Dispositivos, sus estados y configuraciones

Como toda plataforma de manejo de hogares inteligentes, este sitio web ofrece soporte a dispositivos inteligentes relacionados con el hogar. Sin embargo, para el desarrollo de esta versión solo se han seleccionado cinco para desarrollar, los cuales son: lámpara, aspiradora, aspersor, aire acondicionado y horno.

Todos los dispositivos cuentan con dos visualizaciones, una simple (tile) y otra avanzada.

En cuanto a las simples, son las que nos encontramos en las secciones principales. Estas se encuentran compuestas por un ícono representativo del tipo de dispositivo en cuestión, su nombre y la posibilidad de que el usuario pueda prender y apagar.

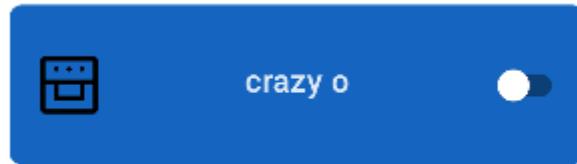


Imagen 1.2.1 Ejemplo de tile simple, en este caso el horno.

Para el caso de querer acceder a las visualizaciones avanzadas de cada dispositivo, solamente tendremos que presionar su tile correspondiente. Estas cuentan con una serie de botones, sliders y otros componentes que permiten una configuración avanzada del dispositivo tales como la temperatura del horno, la intensidad del aire acondicionado, o el color de la luz etc.

A continuación se detalla cada dispositivo.

1.2.1 Luz

La vista avanzada de este dispositivo resulta ser muy similar a la del prototipo final, donde podemos seleccionar un color, como así también regular su intensidad (o brillo). Para esta última tenemos dos posibilidades, asignarla mediante un slider o directamente ingresar por texto el nivel de porcentaje de intensidad que deseamos. Para el caso del color, podemos acceder a una nueva vista donde nos encontraremos con un picker.

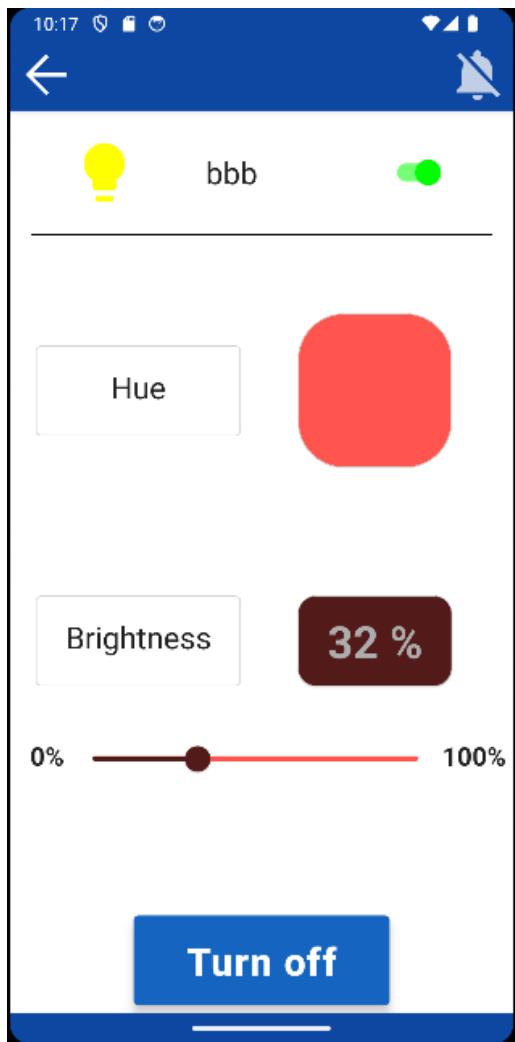


Imagen 1.2.1.1 Vista principal de la configuración avanzada de la luz.

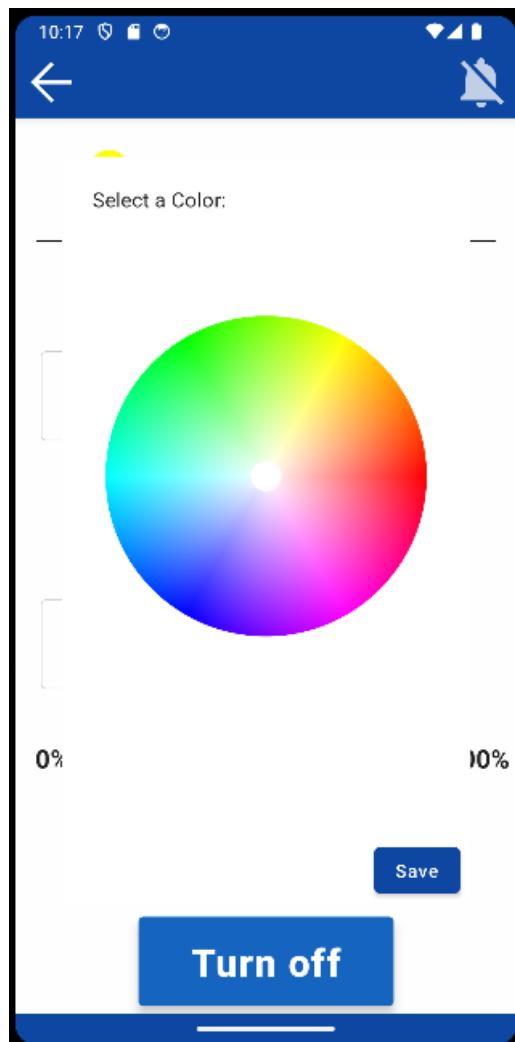


Imagen 1.2.1.2 Vista con el color picker.

1.2.2 Horno

En la configuración avanzada del horno se pueden modificar la temperatura, la fuente de calor, el nivel de circulación del aire (convección) y el modo parrilla. La temperatura se modifica a través de dos botones que aumentan o reducen la temperatura de a cinco grados centígrados y estos se pueden establecer entre 180 y 230 grados centígrados. La fuente de calor tiene tres modos, arriba, abajo y convencional, los cuales se seleccionan mediante los iconos que se encuentran a la derecha de la leyenda “heat source”. La circulación del aire

viene dada en tres modos: convección, convección económica y apagado, que se encuentran a la derecha de la leyenda “convection”. Por último, el modo grill se puede prender, prender en modo económico y apagar, y se encuentra a la derecha de su respectiva leyenda.

El diseño de esta vista también es bastante fiel a la del prototipo final y tiene un gran parecido con el de la página web. Hasta se reutilizan la mayoría de los logos. Cabe destacar que esto último sucede en varias vistas de la aplicación

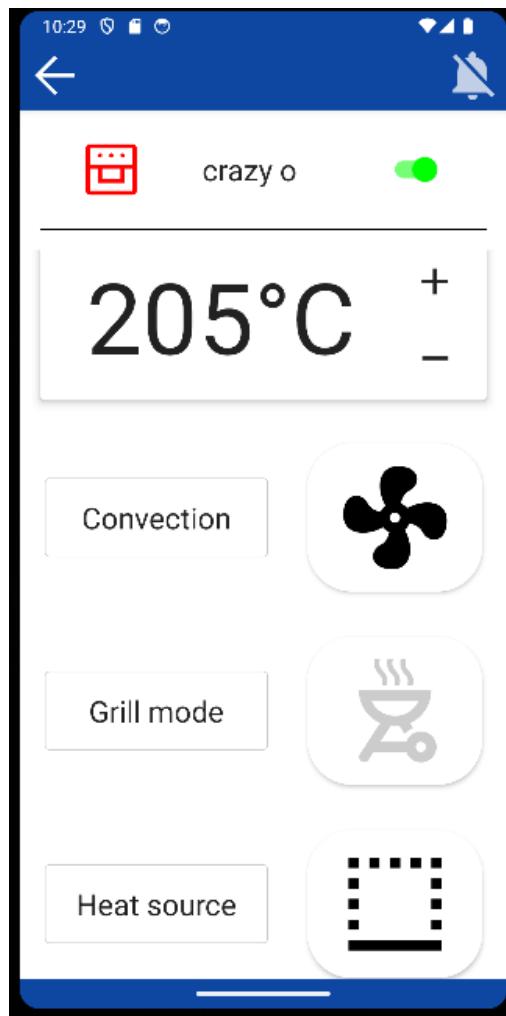


Imagen 1.2.2.1 Vista de la configuración avanzada del horno.

1.2.3 Aspiradora

En el caso de este dispositivo se pueden modificar el modo de uso que son trapear y aspirar, y un tercero que le indica al dispositivo que debe acercarse a su estación de carga para recargar su batería. Luego, se puede seleccionar qué habitación se desea limpiar a través

de una lista desplegable. Por último, se muestra el porcentaje de carga, ya que este al bajar del 5% interrumpe sus tareas para volver automáticamente a su estación de carga

Por otro lado, en la tile simple, al estar la aspiradora prendida, se muestra el estado de carga de su batería.

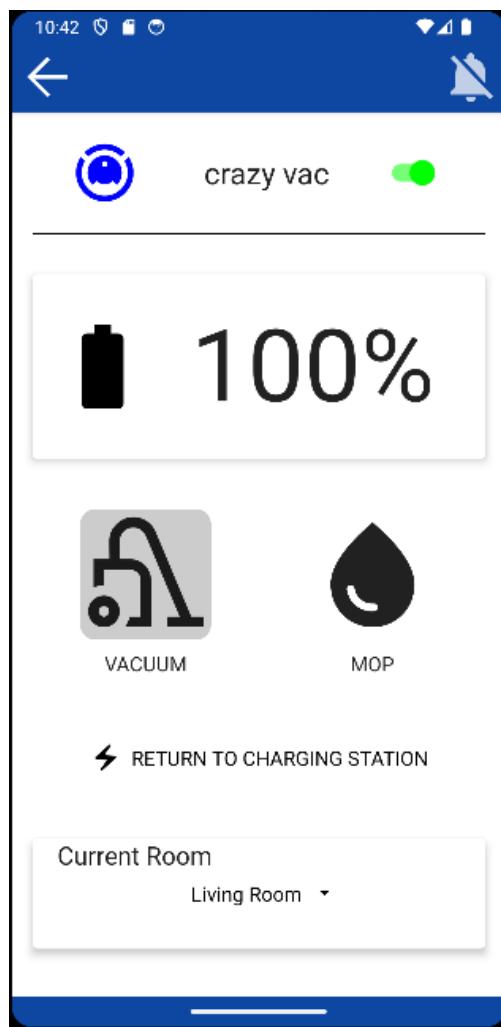


Imagen 1.2.3.1 Vista de la configuración avanzada de la aspiradora

1.2.4 Aire Acondicionado

En la configuración avanzada del aire acondicionado se pueden modificar cinco aspectos del aire acondicionado. La primera es la temperatura (limitada entre 18 y 38 grados centígrados), que se puede modificar o a través de dos botones para aumentar o reducirla en una unidad. La segunda, es el modo del aire acondicionado, frío, caliente o ventilador, que se modifican a través de un *carrousel compuesto por tres botones*, de los cuales solo uno de los

tres puede estar seleccionado a la vez. Tercera, a través de un par de botones se modifica el desplazamiento de las aspas verticales del aire acondicionado, pudiendo elegir uno de los cuatro ángulos disponibles (22, 45, 67, 90 grados) o que varíe automáticamente. La cuarta función idéntica en cuanto aspecto a la tercera, permite modificar el desplazamiento de las aspas horizontales. Por último, se puede establecer la velocidad del ventilador, siendo estos 25, 50, 75 y 100% de su potencia, o dejarlo en automático.

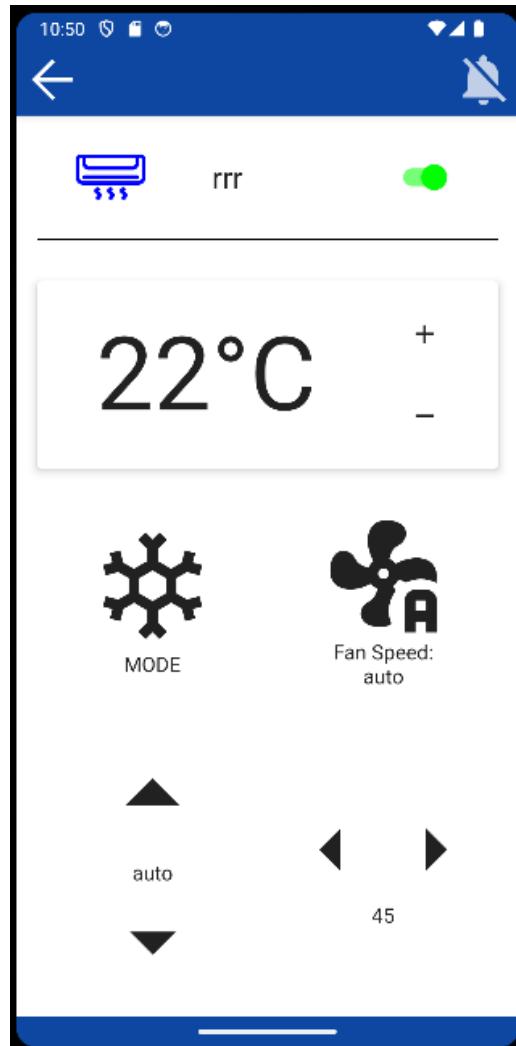


Imagen 1.2.4.1 Vista de la configuración avanzada del aire acondicionado.

1.2.5 Aspersor

En la vista avanzada del aspersor se puede configurar la cantidad de agua a dispensar que corresponde a un valor entre 0 y 100, junto a una unidad de volumen que podemos seleccionar a través de una lista desplegable. Podemos utilizar esta funcionalidad siempre y

cuando el dispositivo se encuentre apagado. Una vez encendido, se abre una nueva vista y se muestra la cantidad de agua que va siendo dispensada junto a la opción de frenar la acción y apagar el dispositivo. La otra opción que se ofrece es la de simplemente encender o abrir el aspersor y dejar que este disperse agua indefinidamente. Esta última, al momento de ejercerse, también se despliega en una nueva vista.

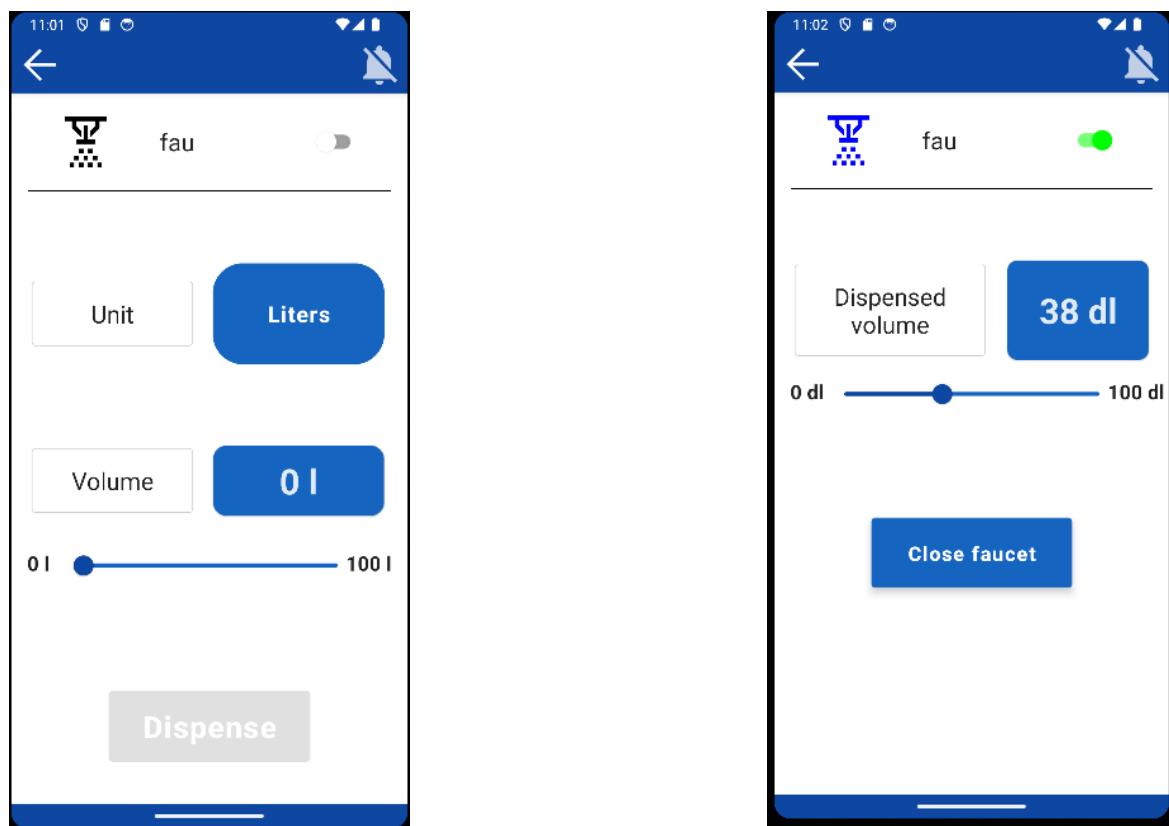




Imagen 1.2.5.1 Vistas de la configuracion del aspersor

1.3 Consultar las rutinas

Las rutinas permiten programar una serie de acciones que se ejecutan automáticamente en un dispositivo electrónico, o en un conjunto de dispositivos. Para el caso de la implementación de esta aplicación solo era un requisito funcional obligatorio poder consultarlas y ejecutarlas. Como se mostró al principio, en la sección de navegación, estas cuentan con su propia vista y también se despliegan mediante tiles. A diferencia de los dispositivos, estos tiles solo contienen el nombre de la misma y un botón que permite accionarlas. Además, no hay ningún tipo de filtrado.



Imagen 1.3.1 Tile correspondiente a la rutina “Bed”.

Si pulsamos sobre el tile se abrirá una nueva vista con la información correspondiente a la rutina, es decir, las acciones a realizar por cada uno de los dispositivos involucrados.

Finalmente, también se mostrará otro botón para poder ejecutar tal rutina.

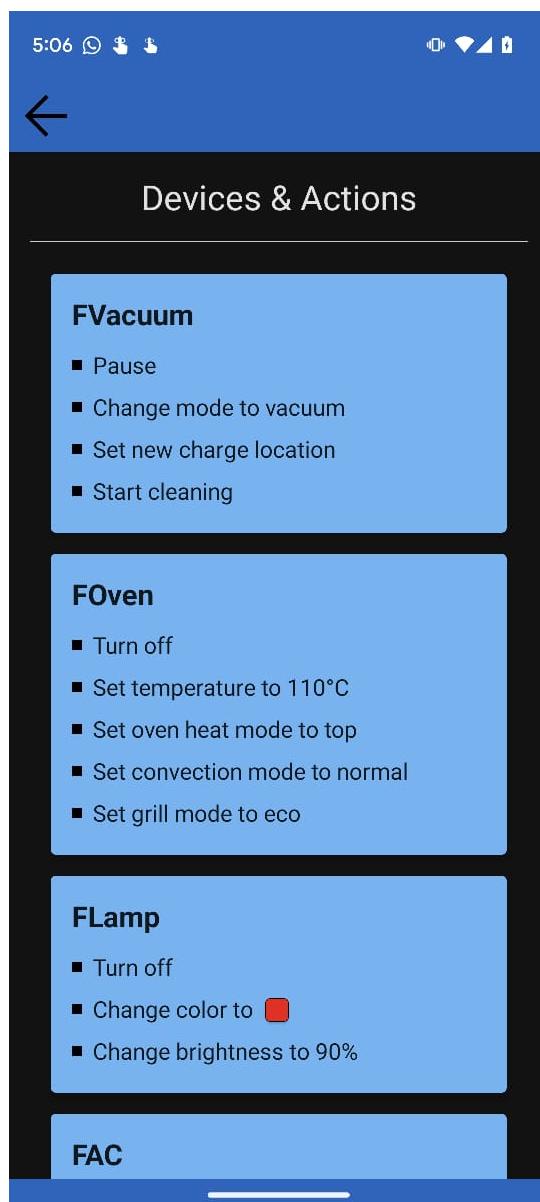


Imagen 1.3.2 Vista avanzada de una rutina.

1.4 Notificaciones

La implementación de un sistema de notificaciones fue de las novedades más importantes respecto a la entrega anterior ya que resulta ser una propiedad bastante intrínseca de los dispositivos móviles. A continuación se explica brevemente el flujo de funcionamiento de estas.

Dentro de cada pantalla de configuración avanzada de dispositivo, en la barra superior, nos encontraremos con un ícono de campana que representará el estado de las notificaciones, es decir, si están encendidas o no para tal dispositivo. Siempre comenzarán apagadas.



Imagen 1.4.1 Barra superior de cada vista de configuración avanzada.

Dado el caso de que deseemos activar las notificaciones de un determinado dispositivo, por ejemplo, la luz, aquellas modificaciones que fuesen realizadas sea desde la API o desde la aplicación web, para este caso serían cambiar el color o brillo, emitirían un mensaje como el siguiente a nuestro celular.

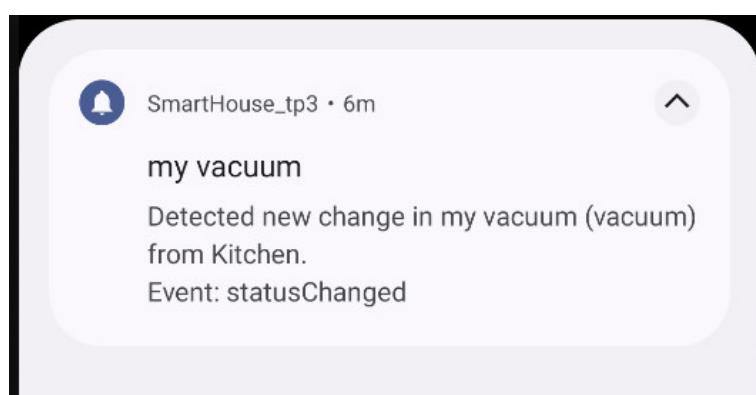


Imagen 1.4.2 Ejemplo de notificación.

Si presionamos esta ventana, nos redirige a la vista de devices.

2. Requisitos no funcionales implementados

En este apartado se explicarán las decisiones tomadas en cuanto al desarrollo de algunas de las posibilidades y desafíos particulares que se plantean al momento de implementar una aplicación para Android o dispositivos móviles

2.1 Idiomas

La aplicación ha sido diseñada para brindar soporte tanto en español como en inglés, con el fin de alcanzar a una amplia audiencia global. Para lograr esta funcionalidad multilingüe, hemos aprovechado las facilidades que ofrece Android Studio. Gracias a su entorno de desarrollo integrado, hemos utilizado herramientas como el sistema de recursos de Android, que permite la traducción de cadenas de texto y la gestión de diferentes archivos de recursos para cada idioma. Además, Android Studio cuenta con una interfaz intuitiva que facilita la creación y edición de los archivos de traducción, lo que ha permitido agilizar el proceso de localización y adaptación de la aplicación a diferentes idiomas. Gracias a estas facilidades, hemos logrado proporcionar una experiencia fluida y personalizada a los usuarios, independientemente de su idioma preferido. En anexo se pueden encontrar algunas vistas de ejemplo con el idioma español.

2.2 Personalización

La aplicación ofrece a los usuarios la opción de personalizar su experiencia a través del modo oscuro. Esta funcionalidad permite cambiar la apariencia de la interfaz de usuario a tonos más oscuros, lo que resulta especialmente útil en entornos de poca luz o para aquellos usuarios que prefieren una estética más discreta.

La implementación del modo oscuro en la aplicación se simplifica gracias a las herramientas proporcionadas por Android Studio. Utilizamos el sistema de recursos de Android, que nos permite definir diferentes conjuntos de recursos para los modos claro y oscuro. A través de los archivos XML correspondientes, pudimos ajustar los colores y estilos

de la interfaz en función del modo seleccionado. En el anexo se pueden encontrar capturas de las diferentes vistas de la app en modo oscuro.

Por otro lado, tenemos la sección de favoritos la cual le permite a los usuarios seleccionar los dispositivos de uso más frecuente. Podemos marcar y desmarcar un dispositivo como favorito a través de su respectiva pantalla de configuración avanzada. Si miramos la barra superior:



Imagen 2.2.1 Barra superior de cada vista de configuración avanzada. Se observa que el dispositivo está marcado como favorito.

Finalmente, destacamos también como opción de personalización a las notificaciones, las cuales ya fueron explicadas en su respectiva sección anteriormente.

2.3 Factor de Forma

El emulador de Android Studio desempeñó un papel fundamental en el desarrollo de nuestra aplicación, ya que nos brindó la capacidad de probarla en diferentes tamaños de pantalla, incluyendo tablets. Esta funcionalidad nos permitió realizar adaptaciones específicas para optimizar la experiencia de usuario en dispositivos de mayor tamaño. La vista que decidimos adaptar para tablet (tanto en orientación vertical como en horizontal) es la de routines.

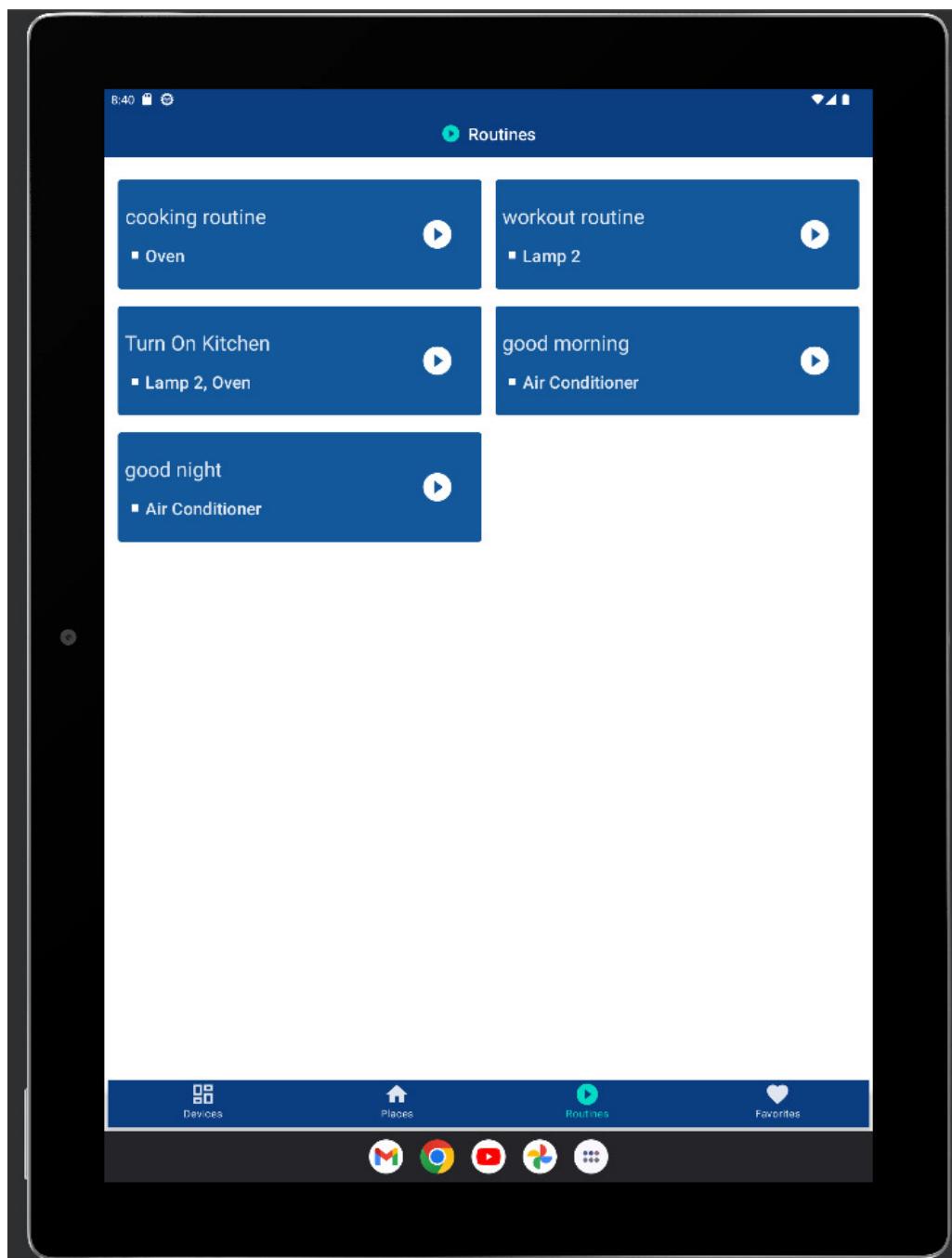


Imagen 2.3.1 Vista routines en orientación vertical para tablet.

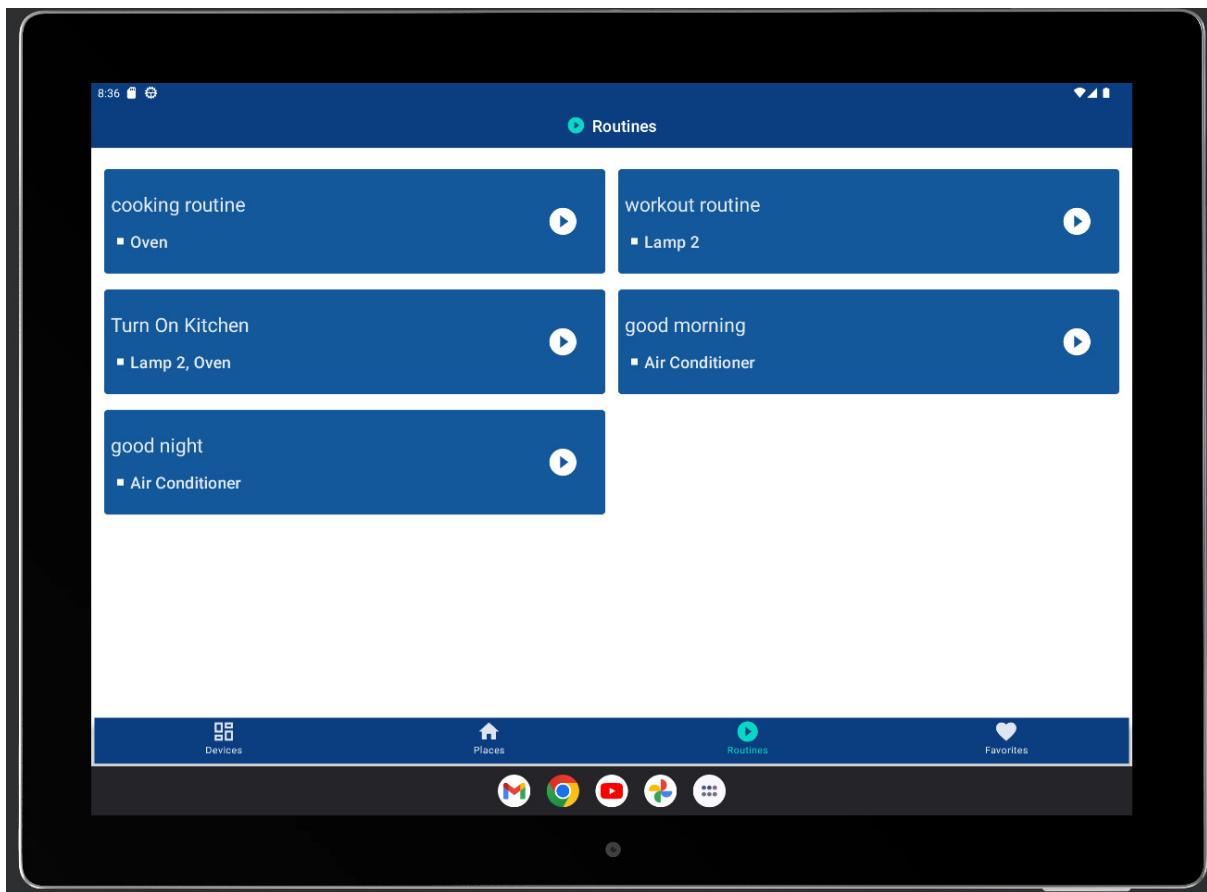


Imagen 2.3.1 Vista routines en orientación horizontal para tablet.

La principal diferencia con las vistas de celular es que estas muestran en cada tile los nombres de los dispositivos asociados a cada rutina.

2.4 Orientación del dispositivo

Siguiendo con la misma línea de la sección anterior, se eligió la vista de rutinas para implementar de forma tal que pueda visualizarse con la orientación horizontal.

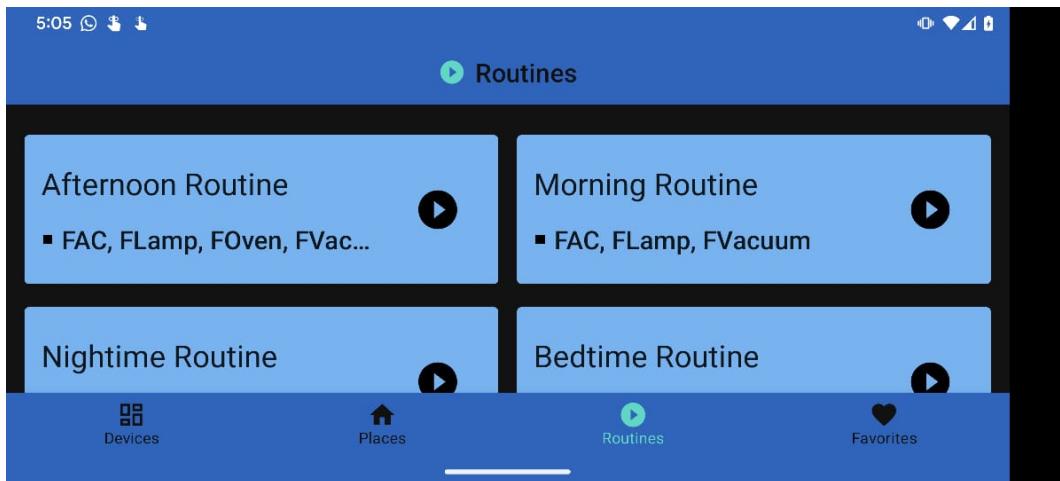


Imagen 2.4.1 Vista Routines en orientación horizontal.

Nuevamente la principal diferencia con la vista de orientación vertical es que la horizontal muestra en cada tile los nombres de los dispositivos asociados a cada rutina.

2.5 Funcionar en dispositivos Pie 9.0 - API Level 28 (o superior)

Como se puede ver en el archivo build.gradle, el minSdk configurado es 28, pero para poder brindar la mejor experiencia al usuario el targetSdk se mantuvo en el más alto disponible en el momento, 33.

3. Decisiones de usabilidad tomadas durante la etapa de implementación

3.1 Cambios en los prototípos

Si bien nuestro prototipo final, al estar hecho en lápiz, no exhibe ciertos detalles que resultan ser importantes como elección de colores, aun así la implementación final de estos

sugirió realizar cambios en cuanto a la estructura y base de estos prototipos. A continuación, describimos con detalle estos cambios.

Empezando por la luz, como se mencionó anteriormente, este dispositivo no sugirió demasiados cambios respecto al implementado en la aplicación, lo más importante a resaltar es que el color lo elegimos en una vista aparte y el valor de intensidad ahora también lo podemos ingresar mediante texto.

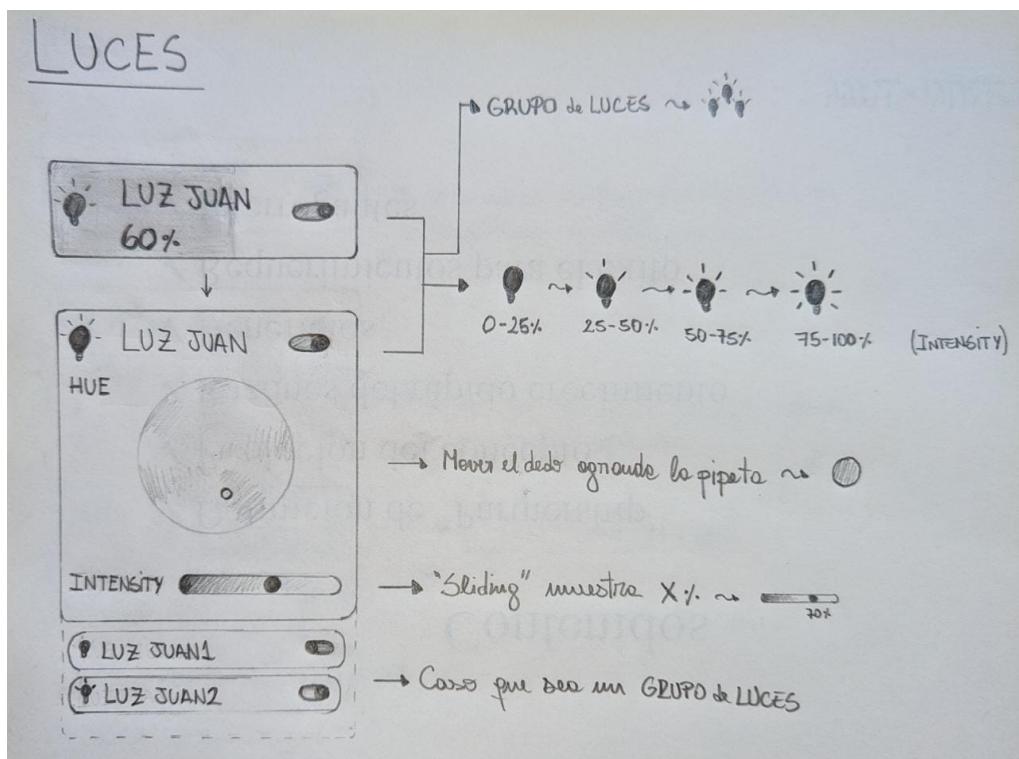


Imagen 3.1.1 Prototipo final de la configuración avanzada de la luz.

En cuanto a nuestra versión del tile, la diferencia principal es que no muestra el valor del porcentaje de intensidad.



Imagen 3.1.2 Tile de la luz.

Otra diferencia importante con el prototipo es que, como se puede apreciar en la imagen, este posee el caso de grupo de luces. Esto fue descartado de nuestra implementación por cuestión de tiempo y prioridades.

Algunos detalles que valen la pena mencionar de la vista implementada son que el color de fondo del field que muestra el porcentaje de intensidad, como así también del slider, varían a medida que vamos cambiando este valor. Además, si nuestra luz se encuentra apagada, el color de fondo del slider pasará a ser gris. Estos cambios le permiten al usuario tener una referencia visual clara y directa sobre el nivel de luminosidad seleccionado.



Imagen 3.1.3 Correspondencia de colores entre el slider y el text field.

Imagen 3.1.4 Slider al estar la luz apagada.

Siguiendo ahora por el horno, la principal diferencia que encontramos con el prototipo final es la organización de cada componente que conforma a la vista, pues el sistema de carrusel se mantiene y las funcionalidades implementadas son todas las mismas. Solo que ahora, como se vio anteriormente, la disposición de los elementos en pantalla se da en forma de una sola columna y varias filas.

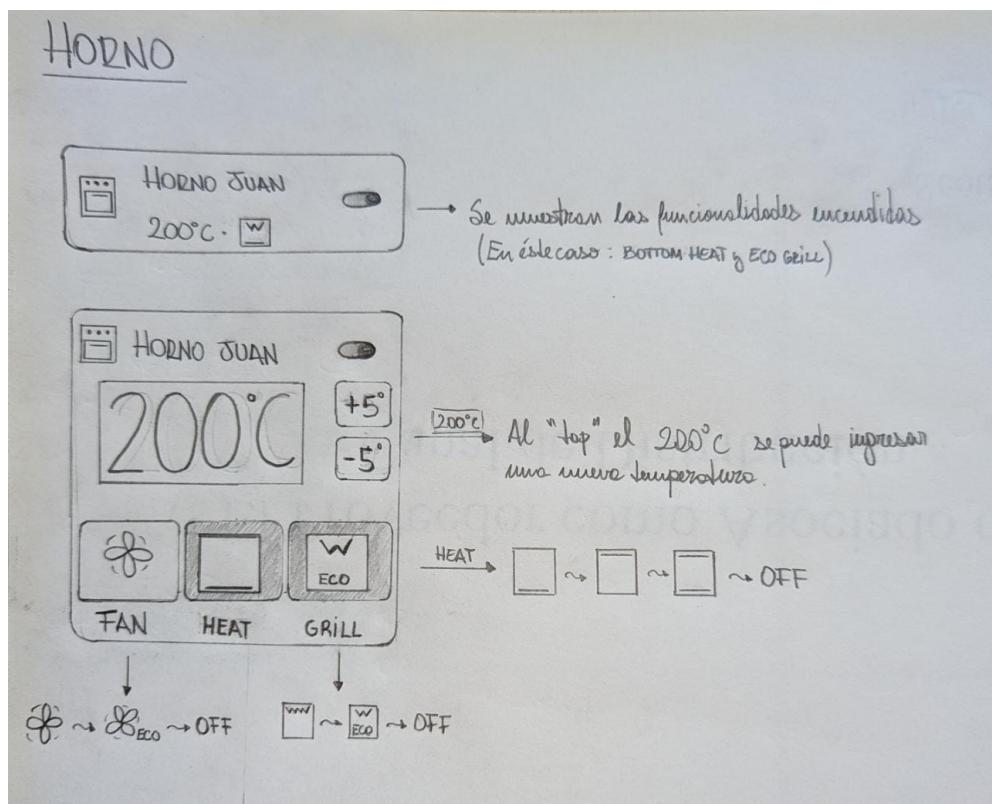


Imagen 3.1.5 Prototipo final del horno

Al igual que la luz, el tile del horno tampoco muestra un estado.

Unos detalles a mencionar sobre la versión que se encuentra en la app son que, por ejemplo, para representar las opciones “eco” de aquellos modos que la poseen se pintaron los respectivos iconos de verde para que el usuario pueda rápidamente entender y asociar la función de tal opción. Además, cuando el dispositivo se encuentra apagado, los iconos de todos los modos se pintan de un color gris claro. Esto ayuda a los usuarios a hacer una rápida diferenciación visual de los estados.



Imagen 3.1.6 Logo eco grill



Imagen 3.1.7 Logo de modo fan con el dispositivo apagado

Siguiendo con otro dispositivo, si nos vamos al caso del aspersor, tenemos que este fue el que más cambios sufrió respecto de la versión prototipada. La principal razón fue porque teníamos pensado representar la cantidad de volumen de agua a dispensar a través de una especie de cilindro interactivo que íbamos a poder controlar mediante un desplazamiento hacia arriba o abajo. Dada la poca experiencia que tenemos con jetpack Compose, tuvimos que optar por una opción más sencilla, pero no menos práctica, para poder implementar esta funcionalidad. Sin embargo, la versión prototipada no contempla del todo la opción de dispersar una cantidad de agua indeterminada, por lo que fue necesario hacer los agregados que se mostraron anteriormente para generar una mayor retroalimentación. Además, de esta forma el usuario también tiene un mayor control de las operaciones que puede realizar.

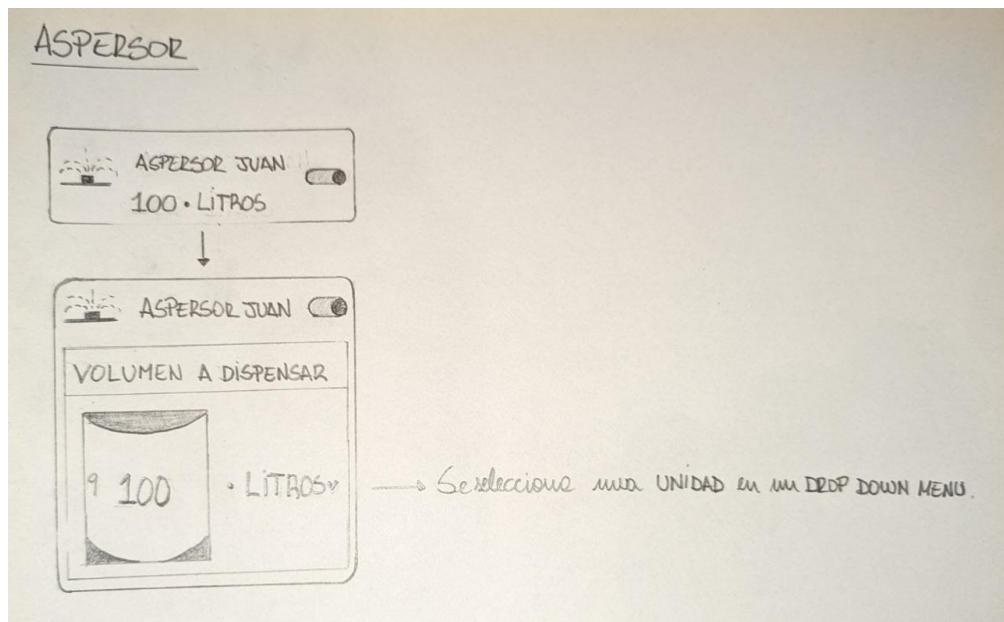


Imagen 3.1.8 Prototipo final del aspersor.

En cuanto a la aspiradora, no sufrió muchos cambios respecto al prototipo, solo un poco el cómo se desplegó cada componente.

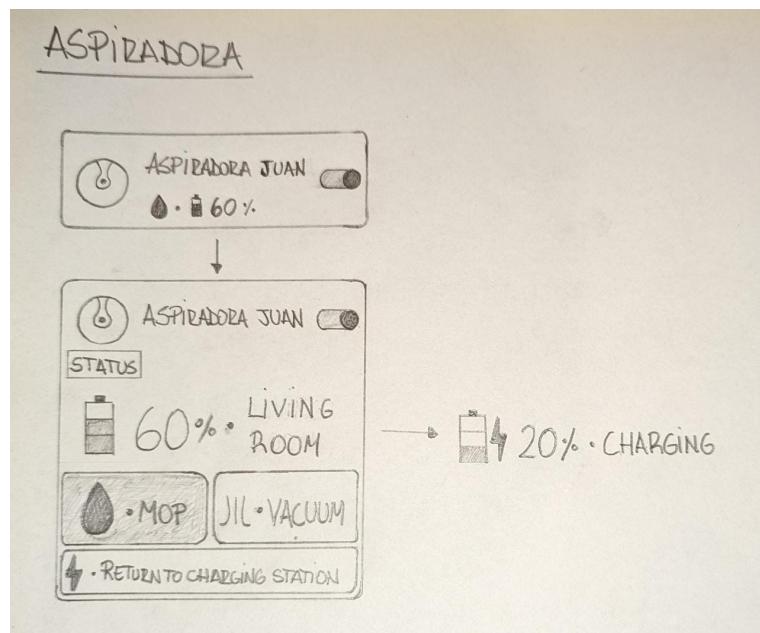


Imagen 3.1.9 Prototipo final de la aspiradora.

Lo mismo ocurre para el aire acondicionado.

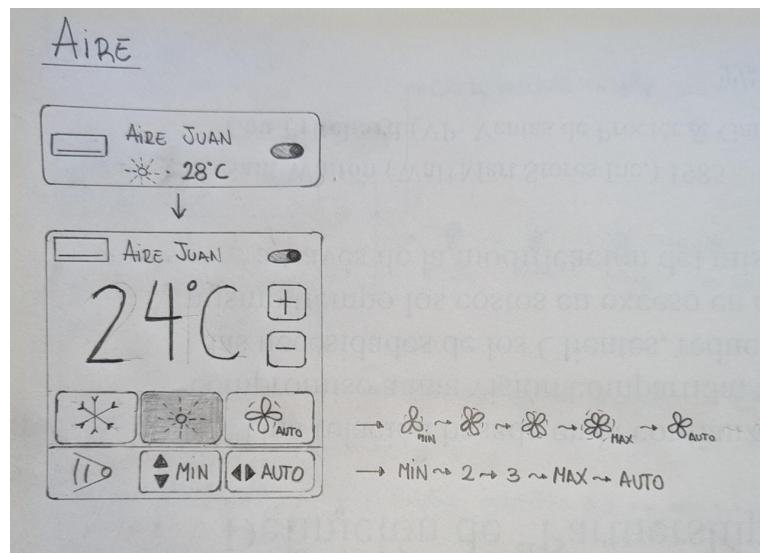


Imagen 3.1.10 Prototipo final del aire acondicionado.

Dejando de lado la configuración de cada dispositivo en particular y volviendo a la navegación de la aplicación, si comparamos el resultado final con el prototipo final tenemos que lo obtenido resulta ser bastante acercado a lo concluido durante la primera entrega.

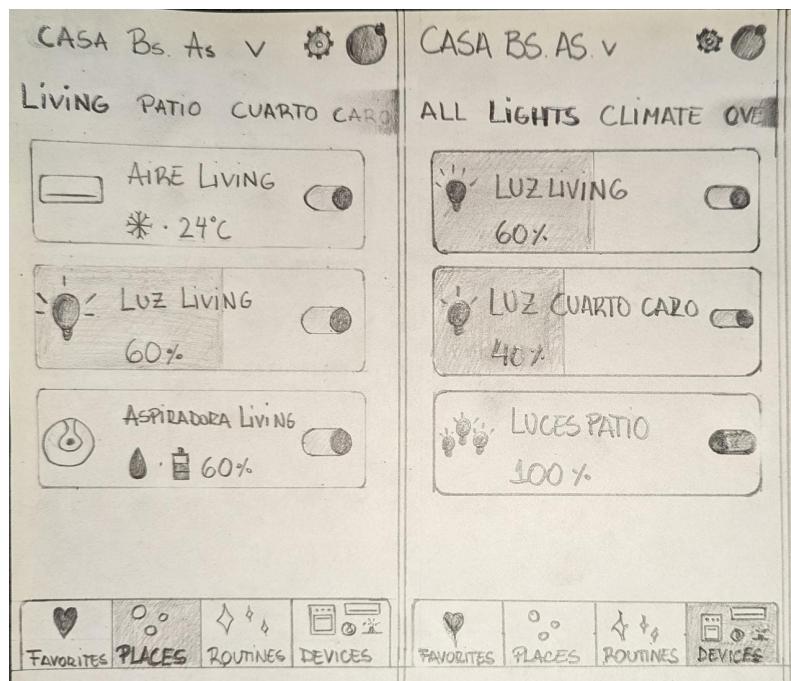


Imagen 3.1.9 Prototipo final de las principales vistas de la aplicación móvil.

Como ya vimos anteriormente, la barra de navegación contiene las mismas secciones con la diferencia de el orden e iconos. También fue mencionado que los tiles, en nuestra versión final, no poseen en su mayoría iconos con el estado del dispositivo que representan. Sin embargo, el mismo hecho de respetar el uso del mismo tipo de tiles se mantiene y la disposición de carruseles para filtrar los dispositivos por tipo y ambiente también se respeta.

3.2 Colores

El cambio más importante a destacar con respecto a la paleta de colores yace en que fue cambiada de la utilizada para la página web. Esto fue así debido a que una de las críticas que se recibió en dicha entrega fue en que si bien la aplicación web tenía un enfoque minimalista, abusaba demasiado de esto y podría llegar a parecer monótona o hasta no finalizada. Además el no categorizar diferentes secciones por algún color en particular podía llegar a aburrir al usuario.

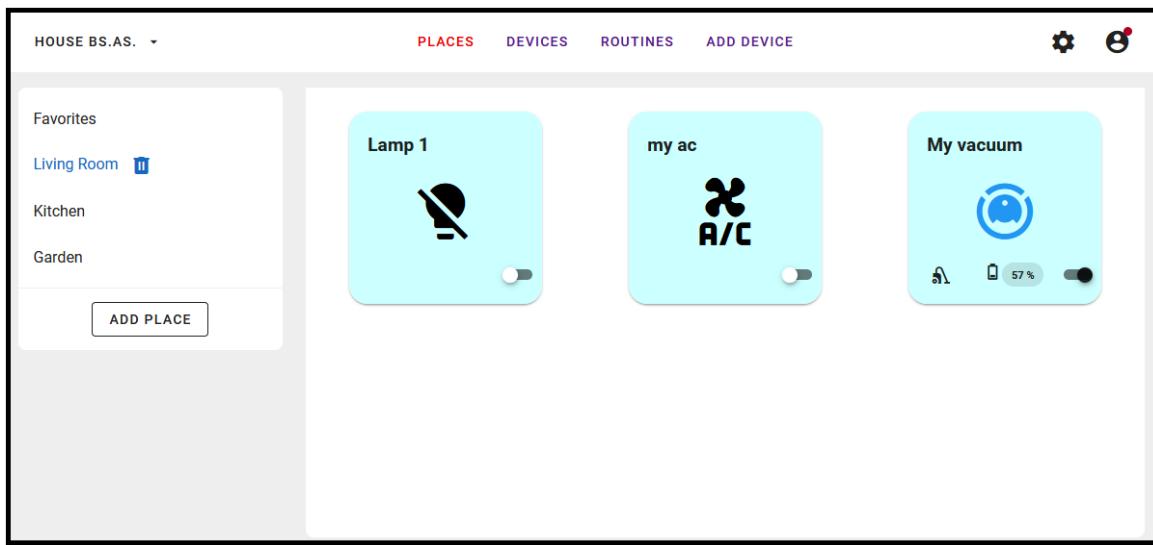


Imagen 3.2.1 Una de las vistas principales del sitio web.

Ahora, al combinar blanco con un azul mas oscuro y negro, con diferentes pequeños agregados, se genera una buena distincion entre las secciones de cada vista de la app. Esto a su vez otorga una mayor variedad visual y le da al usuario una mejor legibilidad y enfoque.

3.3 Globalización

Lo único a mencionar respecto a este aspecto es que, como ya se dijo anteriormente, android studio nos facilitó el hecho de poder soportar varios idiomas. En nuestro caso, la aplicación está preparada tanto para inglés como para español. Donde el inglés es el lenguaje más hablado en el mundo y español uno de los más hablados.

3.4 Feedback

Si bien en esta entrega no hay tantas operaciones que requieran de un buen feedback, como sí pasaba en la anterior, se corrigió una crítica realizada por la cátedra en cuanto a la implementación de la página web. Esta consiste en que cuando no teníamos algún tipo de dispositivo o no teníamos ninguna rutina, solo se mostraba una pantalla vacía. Es decir, no había nada que le indique al usuario que, por ejemplo, no contaba con rutinas y que podía agregar una. Mismo para los tipos de dispositivos y ambientes.

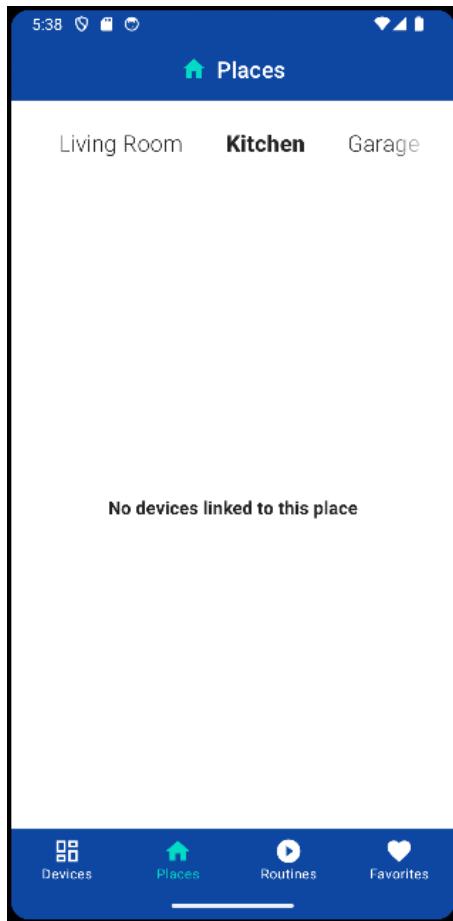


Imagen 3.4.1 Ejemplo de feedback con ambiente Kitchen.

Como ya se mencionó anteriormente, otro aspecto mejorado fue el de la funcionalidad de que el aspersor pueda dispensar una cantidad de agua indeterminada. En el caso de la página web, podía pasar de que el usuario abra el aspersor, intente dispensar y se encuentre con que por algún motivo no puede. Ahora, si abre el aspersor se despliega la siguiente vista:

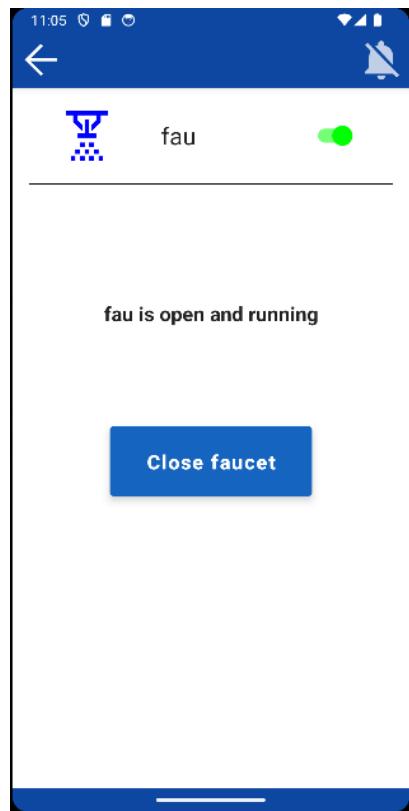


Imagen 3.4.2 Aspersor dispensando indefinidamente.

Esto le ayuda al usuario a concluir rápidamente que si desea utilizar otra función, como la de dispensar una cantidad de volumen determinada, debe primero cerrar el aspersor.

Por último, al momento de accionar una rutina, se nos abre un dialogo indicando que esta se pudo ejecutar con satisfacción.

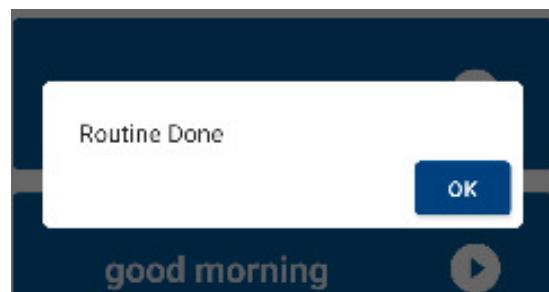


Imagen 3.4.3 Dialog de routine.

4. Limitaciones

A continuación se listan las limitaciones de la implementación de la aplicación:

- El desarrollo del sistema de notificaciones fue una de las partes más complicadas, ya que fue el último en ser agregado a la aplicación. Por ejemplo, como se mencionó anteriormente, todos los dispositivos empiezan sin favoritos ni con notificaciones. Al activarlos, estos quedarán guardados siempre y cuando la aplicación no se cierre. Al cerrar la aplicación, todos los dispositivos vuelven a su estado original, sin favoritos ni con notificaciones.
- BottomBarNavigation: No se pudo modificar el padding entre dispositivos para la tablet. Se deseaba tener los iconos más juntos en el medio de la pantalla como los iconos del resto de las aplicaciones que se observan más abajo.
- Facuet: No se logró desarrollar de manera óptima la clase facuet, el dispense generó más problemas de lo esperado. Asimismo, se tomó la decisión errónea de mostrar una carga de dispense dinámica y para eso se realizan fetches de manera consecutiva. El problema surge de la gran cantidad de api fetch que se realizan para mostrar el progreso de la operación dispense. Lo que se cree que ocurre al finalizar la operación de dispensar es que el uiState logra actualizarse adecuadamente, pero el último api fetch retorna después de esta actualización, pisando los valores reales del uiState. Esto lleva a un estado inconsistente entre la aplicación y la api que se puede ver durante el uso de la aplicación.

5. Archivos necesarios para construir el instalador de la aplicación

Cómo estamos utilizando Android Studio, éste nos permite desarrollar sin tener que estar preocupándonos todo el tiempo sobre el soporte a los distintos dispositivos que nuestros usuarios representativos podrían llegar a utilizar.

Adjuntamos un archivo en formato ZIP con el código fuente de la aplicación. Para que la aplicación pueda comunicarse correctamente con el API, es requisito necesario obtener la IP de la máquina en la qué este esté corriendo. Esto se puede conocer con el comando ipconfig

(Windows) o ip address (Linux). Esta se deberá colocar en el archivo AppConfig.kt, reemplazando “localhost” de la variable API_BASE_URL. De ser necesario, también se debe agregar el archivo local.properties con el path correspondiente a la ubicación del Android SDK.

Adicionalmente, AppConfig.kt contiene constantes que permiten configurar un delay simulado al acceso del API, tal de poder hacer mejor apreciación de los indicadores de carga.

Para construir el APK desde Android Studio, en el menú superior de la ventana, navegamos a Build → Build Bundle(s) / APK(s) → Build APK(s). Esto generará un archivo .apk que debe ser copiado al dispositivo en el que se quiera realizar la instalación.

6. Instructivo de instalación de la APK

Una vez que tengamos el archivo .apk entre los archivos de nuestro dispositivo Android, basta con utilizar un explorador de archivos, encontrar el archivo apk, y abrirlo. Android reconocerá el archivo y mostrará un mensaje ofreciendo instalar la aplicación.

7. Conclusión

El proceso de desarrollo de esta aplicación móvil de control de dispositivos de smart home ha sido una experiencia enriquecedora y desafiante. Nos enfrentamos por primera vez a la tarea de crear una interfaz de usuario amigable, sencilla y práctica, que cumpliera con las expectativas de los usuarios y reflejara los principios de usabilidad y diseño establecidos en nuestra investigación inicial.

Además, tuvimos el desafío adicional de trabajar con tecnologías y lenguajes nuevos para nosotros, como Jetpack Compose y Kotlin. A medida que nos familiarizamos con estas herramientas, enfrentamos las complejidades inherentes a su aprendizaje y aplicación en el desarrollo de la aplicación. Sin embargo, este proceso de aprendizaje nos permitió ampliar nuestros conocimientos y habilidades, y nos llevó a descubrir nuevas formas eficientes de crear interfaces de usuario dinámicas y receptivas.

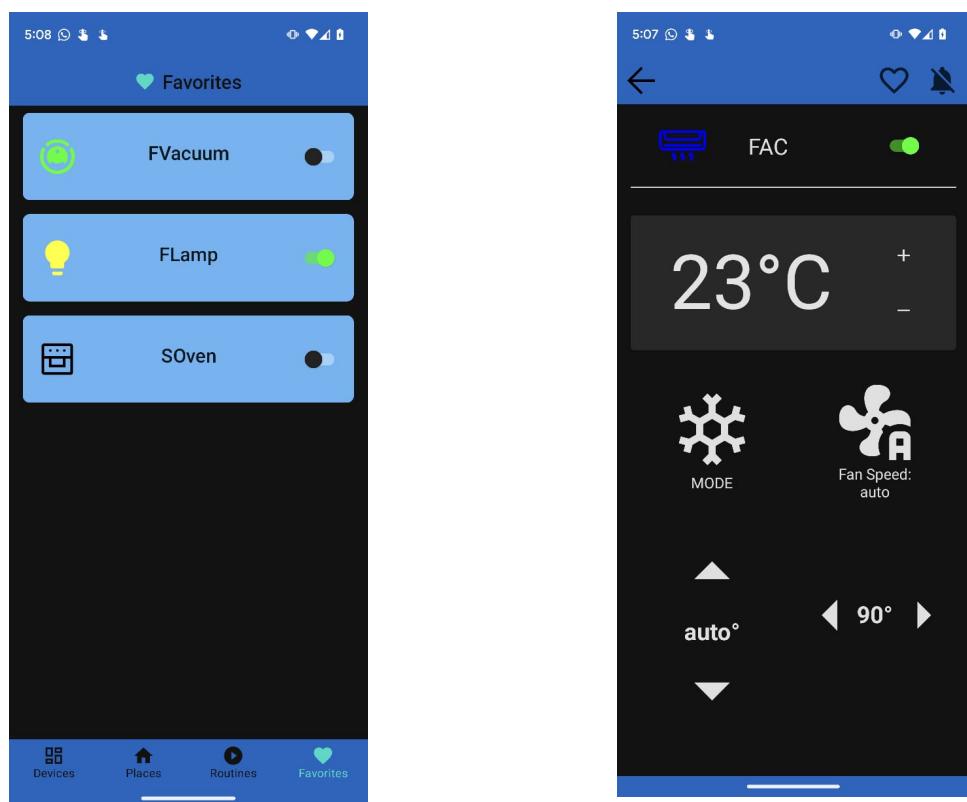
A pesar de los desafíos, el resultado final de todo este trabajo de investigación de usabilidad y diseño de prototipos es sumamente gratificante. Ver cómo nuestras ideas y

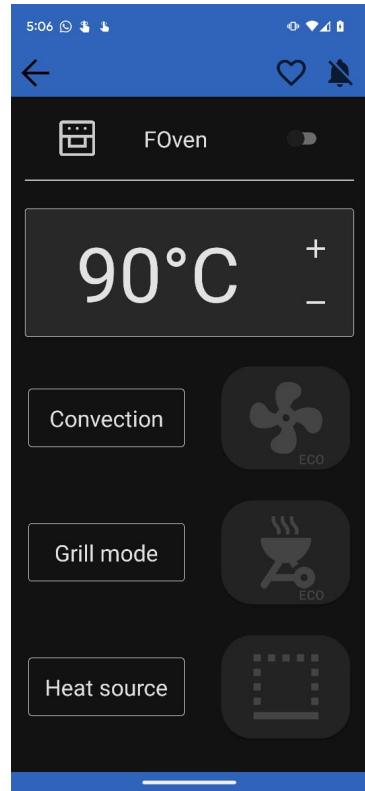
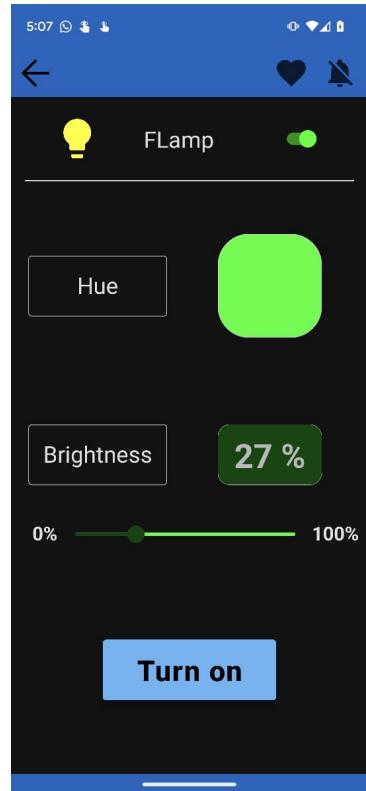
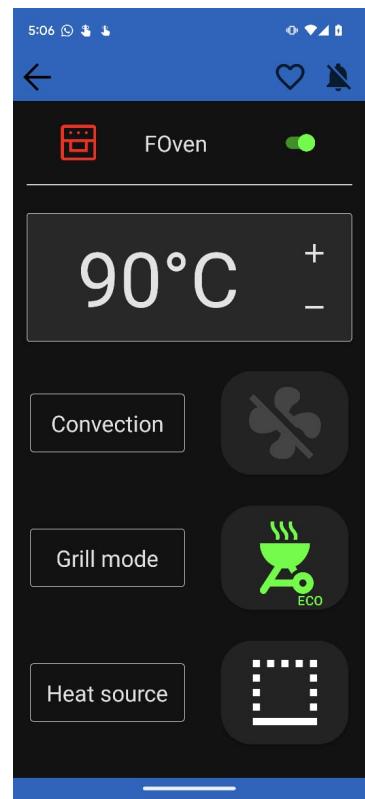
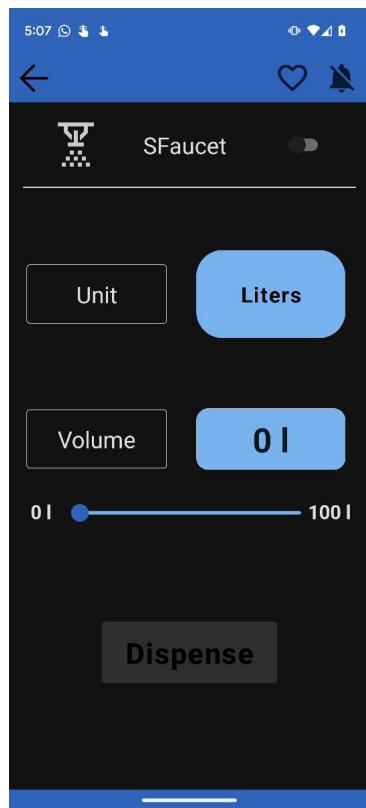
conceptos se materializan en una aplicación funcional y efectiva, que cumple con los requisitos y expectativas de los usuarios, añade un valor significativo a nuestro logro..

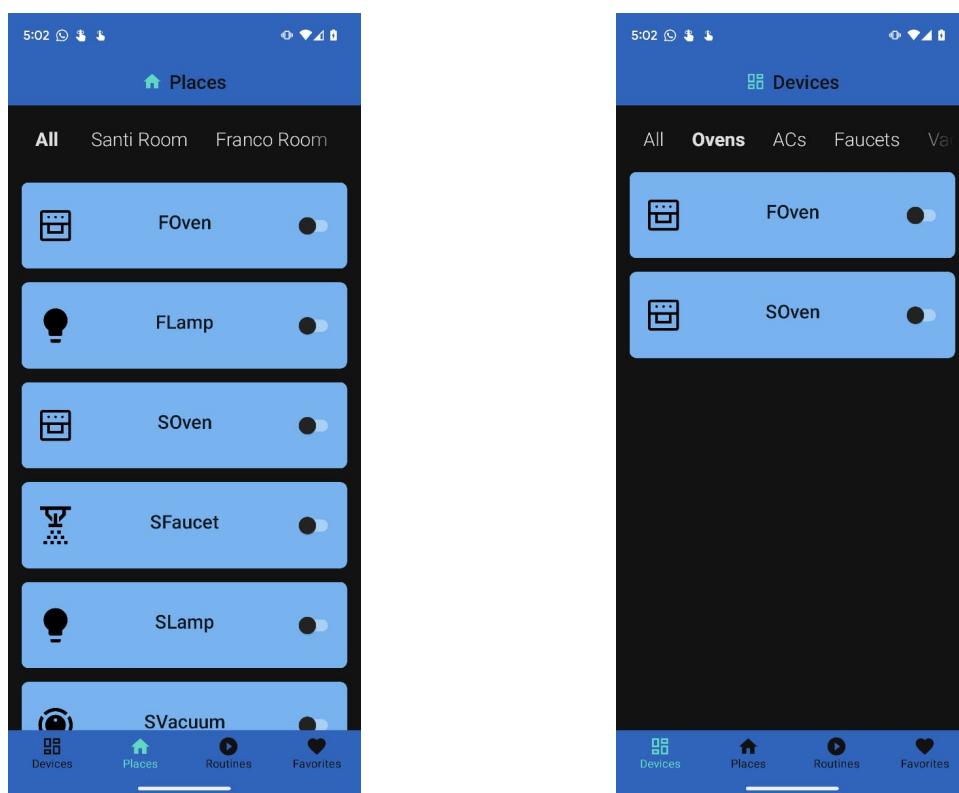
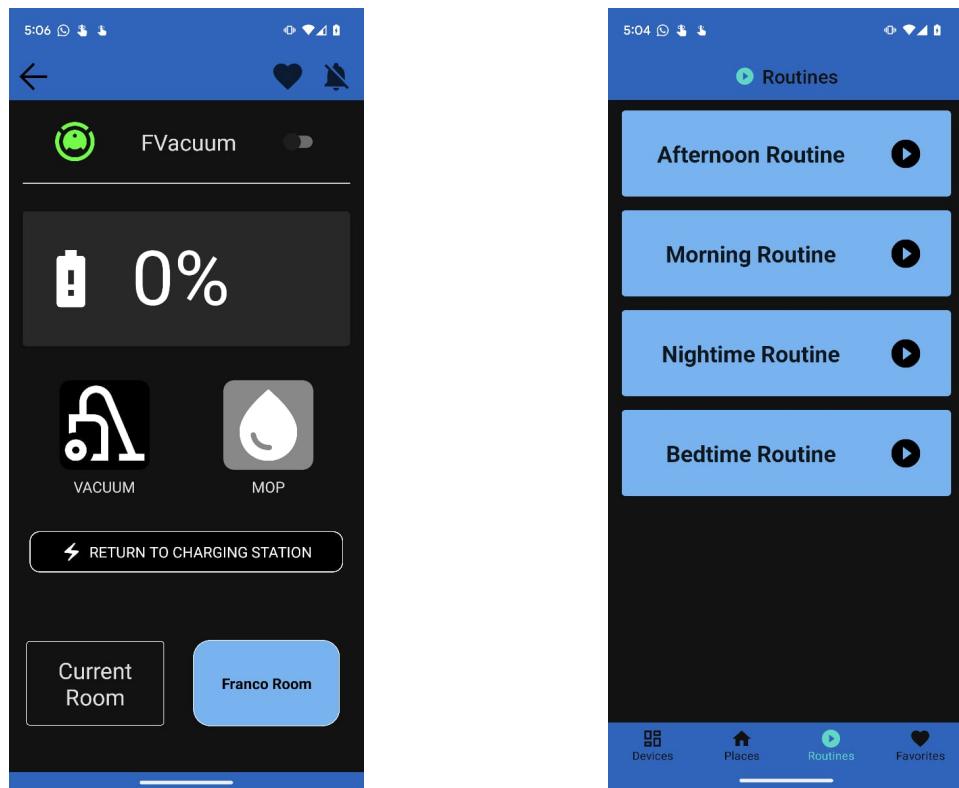
En resumen, el desarrollo de esta aplicación móvil nos ha brindado la oportunidad de enfrentarnos a desafíos técnicos y creativos, así como de aprender y aplicar nuevas tecnologías. A través de este proceso, hemos experimentado el valor y la gratificación de transformar una idea inicial en una realidad funcional, que mejora la vida de los usuarios al proporcionarles un control fácil y eficiente de sus dispositivos de smart home.

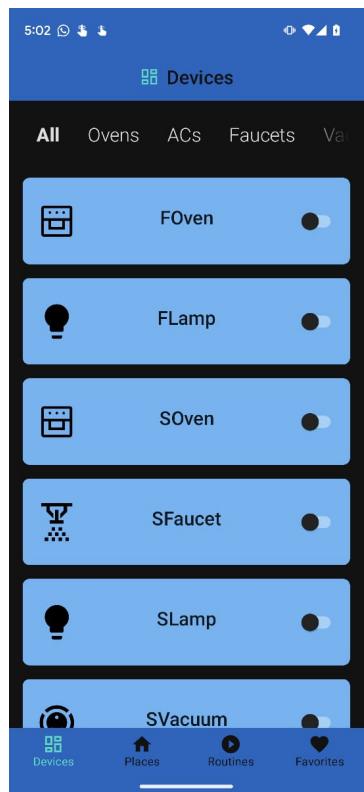
8. Anexo

8.1 Vistas en modo oscuro









8.2 Vistas en idioma español

