

INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA
ESCUELA DE INGENIERÍA Y GESTIÓN

PONTE 2

Android Mobile

AUTOR: Noriega Defferrari, Jose Carlos (Leg. N° 51231)

DOCENTE: Guerrero, Marcela Alejandra

REPOSITORIO: <https://bitbucket.org/itba/ponte2/src>

TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INFORMÁTICA

Lugar: Capital Federal
Fecha: 11/12/2020



Introducción	4
¿Qué es Ponte?	4
Relevamiento	5
Onboarding	5
Iniciar Sesión/Registrarse	7
Perfil	8
Modal de Oferta	9
Voluntariado	10
Registrarse como institución	10
Pedidos	11
Modal de donación	12
Funcionalidades Web	12
Implementadas	12
No Implementadas	13
Funcionalidades a desarrollar	13
Autenticación	13
Pedidos, ofertas y donaciones	14
Instituciones	15
Requisitos No Funcionales	15
Confiabilidad	15
Simplicidad	15
Flexibilidad	15
Usabilidad	15
Performance	15
Seguridad	16
Extensibilidad	16
Desarrollo y QA	17
Diseño de Flujo y UI	18
Paleta de colores	19
Navegación	20
Home	22
Donar	24
Perfil	25
Instituciones y pedidos	27
Pantalla Login/Registrarme	27
Arquitectura	28
Desarrollo	29
Architecture	30
Repository	32
Foundation y UI	32
Aplicación	33

Login	33
Registrarse	34
Home	36
Pedidos e Instituciones	39
Perfil	39
Donaciones	44
Estados de una donación	47
Estado: CREATED	47
Estado: VOLUNTEER-ASSIGNED	47
Estado: BEING-LOOKED-FOR	48
Estado: DELIVERED-BY-DONOR	49
Estado: PICKED-BY-VOLUNTEER	49
Estado: BEING-TAKEN-TO	50
Estado: DELIVERED o CANCELLED	50
Grafico completo del flujo de estados	51
Casos de Prueba	52
Mejoras Futuras	59
Landing	59
Paginación	59
Seguridad	59
Notifications	59
Geolocalización	59
Donaciones	60
Modo Offline	60
Conclusión	60

Introducción

¿Qué es Ponte?

En la actualidad, hemos visto cómo la tecnología ha permitido democratizar distintos aspectos de nuestras vidas; el arte con YouTube y Spotify, el transporte con Uber y Cabify, la gastronomía con Rappi y Glovo, etc etc. Plataformas que han permitido darle una oportunidad a cada vez más personas de vender sus habilidades y talentos, acercándolos a más consumidores día a día. Ponte democratiza otro aspecto de nuestras vidas: la solidaridad

Basado en el latín *pontem* que significa puente, Ponte es un ideal que consiste en conectar a sectores postergados de nuestra sociedad, con sectores que tienen medios y ganas de ayudar. Muchas ONGs de gran renombre, poseen los medios para acercar donantes a las personas que necesitan ayuda, pero debido a la naturaleza del financiamiento de las ONGs, es muy difícil para las pequeñas tener la visibilidad suficiente para poder potenciar su capacidad de ayuda. De la misma manera, hay muchas personas con ganas de ayudar pero no saben como, se les dificulta el traslado entre otras cosas, entonces se ven desmotivados a ayudar. Aquí es en donde entra Ponte.

La idea fue concebida por Sebastian Veira y Noelia Lopez, egresados de Ingeniería Informática del ITBA. como producto final consiste en una plataforma que permite conectar instituciones que realizan pedidos, usuarios que desean donar y voluntarios que se ofrecen para transportar la donación del usuario a la institución. Una donación por lo general se compone de tres elementos: un pedido, una oferta y un voluntario. Un usuario podría donar sin que haya un pedido realizado, en ese caso la donación se almacenaría en el depósito de Ponte. Los mismos usuarios pueden ofrecerse como voluntarios para llevar donaciones, tanto suyas como de terceros.

La plataforma consiste hasta el momento en una versión web, que permite realizar algunas de las funciones básicas de Ponte que se detallarán más adelante en este informe. Pero el objetivo de esta versión del proyecto, es extender los brazos de Ponte al mundo mobile, donde tendría su mayor alcance. En particular, una aplicación para Android, que permite realizar y trackear donaciones y a su vez, guiar a los voluntarios a sus destinos. Se intentará replicar todas las funcionalidades que expone Ponte Web, en Ponte Mobile

Relevamiento

En esta sección, presentamos relevamiento de lo que ya está desarrollado en Ponte Web, para poder así replicarlo en Ponte Mobile

Onboarding

Al comenzar nos encontramos con la siguiente pantalla.

Colaborar hoy es así de simple



Colaborar hoy es así de simple



Creá una cuenta para comenzar.



Contanos que tenés



Un recolector pasará a

Iniciar Sesión/Registrarse

Cuando un usuario quiere registrarse, puede hacerlo como voluntario o como donante.

Bienvenido:)

Entrar

Unite a la causa :)

Nombre	<input type="text"/>
Apellido	<input type="text"/>
Correo electrónico	<input type="text" value="jose.noriega.defferrari@gmail.com"/>
Contraseña	<input type="password" value="....."/>
Teléfono	<input type="text"/>
Dirección	<input type="text"/>
Donante	<input type="checkbox"/>

Al registrarse aceptas los [términos y condiciones](#)

Regístrate



Perfil

Página que muestra el perfil del usuario con su historial de donaciones y donaciones pendientes.

Jose Noriega

Categoría: Donante
Email: jonorieg@itba.edu.ar
Telefono: 123456789
Direccion: Jose Noriega

[QUIERO SER VOLUNTARIO](#)

Donaciones pendientes Historial de Donaciones

30 caramelos

Estado: En búsqueda de institución

[CANCELAR](#)

[QUIERO SER VOLUNTARIO](#)

Donaciones pendientes Historial de Donaciones

30 caramelos

Institución: Noriega DulceHogar
Estado: En búsqueda de voluntario

[CANCELAR](#)

Las donaciones pueden tener distintos estados (explicados en la sección de funcionalidades)

Donaciones pendientes Historial de Donaciones

30 caramelos

Institución: Noriega DulceHogar
Estado: En búsqueda de voluntario

CANCELAR

Voluntario Responsable: Juan

Donaciones pendientes **Historial de Donaciones**

30 caramelos

Estado: Cancelada por el donante

30 caramelos

Institución: Noriega DulceHogar
Estado: Entregada

Voluntario Responsable: Juan

Modal de Oferta

Formulario para completar las características de la donación. Este modal se puede acceder desde la barra superior en cualquier página de Ponte.

PEDIDOS INSTITUCIONES **DONAR ALGO NUEVO** MI PERFIL SALIR

Jose Noriega

Categoría: Donante
Email: jonorieg@itba.edu.ar
Telefono: 123456789
Direccion: Av. Lacroze

QUIERO SER VOLUNTARIO

Donaciones pendientes

¿Tenés algo para donar ? Contanos un poco más, nosotros lo ubicamos. ✕

¿Cuál es tu nombre?:
Jose Noriega

¿Cuál es tu email?:
jonorieg@itba.edu.ar

¿Cuál es tu número de celular?:
123456789

¿Qué bien tenés para donar?:
Contanos qué tenés para donar.

¿Cuantas unidades podés donar?:
1

¿En qué horario podrias buscar las donaciones?:
¿Dia?: Lunes ¿Hora?: De 8 a 12

+
¿Algún otro comentario?:
Agregá cualquier comentario que desees.

ENVIAR

Voluntariado

Formulario con especificaciones para ser voluntario

Como voluntario nos ayudás a transportar las donaciones. ¡Gracias!

¿Qué tamaño de donación podés transportar?

Chico (ej. una caja de tiza)

¿En qué horario podrían pasar a buscar las donaciones?

¿Día?: Lunes ¿Hora?: De 8 a 12

+
¿Algún otro comentario?:
Agregá cualquier comentario que desees.

ENVIAR

Registrarse como institución

Formulario con preguntas para poder registrarse.

¿Eres una institución?

Una plataforma para que donar todos aquellos que usas y no le encuentres siendo aprovechado de manera

El objetivo es ser el puente entre los sectores más vulnerables y aquellos que los quieren

REGISTRARSE COMO INSTITUCIÓN

Para registrarte como Institución, por favor completa el siguiente formulario y nos comunicaremos con vos.

¿Cuál es el nombre de la institución?:
Dejanos el nombre de la institución.

¿Cuál es el domicilio de la institución?:
Dejanos el domicilio de la institución.

Contanos un poco más sobre la institución:
Contanos un poco más sobre la institución.

¿Cuál es tu nombre?:
Dejanos tu nombre

¿Cuál es tu email?:
Dejanos tu mail

¿Cuál es tu número de celular?:
Dejanos tu número de celular

ENVIAR

El buscador de instituciones permite buscar instituciones por su nombre



Noriega DulceHogar

jose.noriega.deferrari@gmail.com

Jo Noriega Defferra

¡Donaciones de todo tipo!
Avenida Congreso 2077, 4B
01168950088

Pedidos

Permite ver y buscar pedidos hechos por instituciones y hacer donaciones



30 caramelos

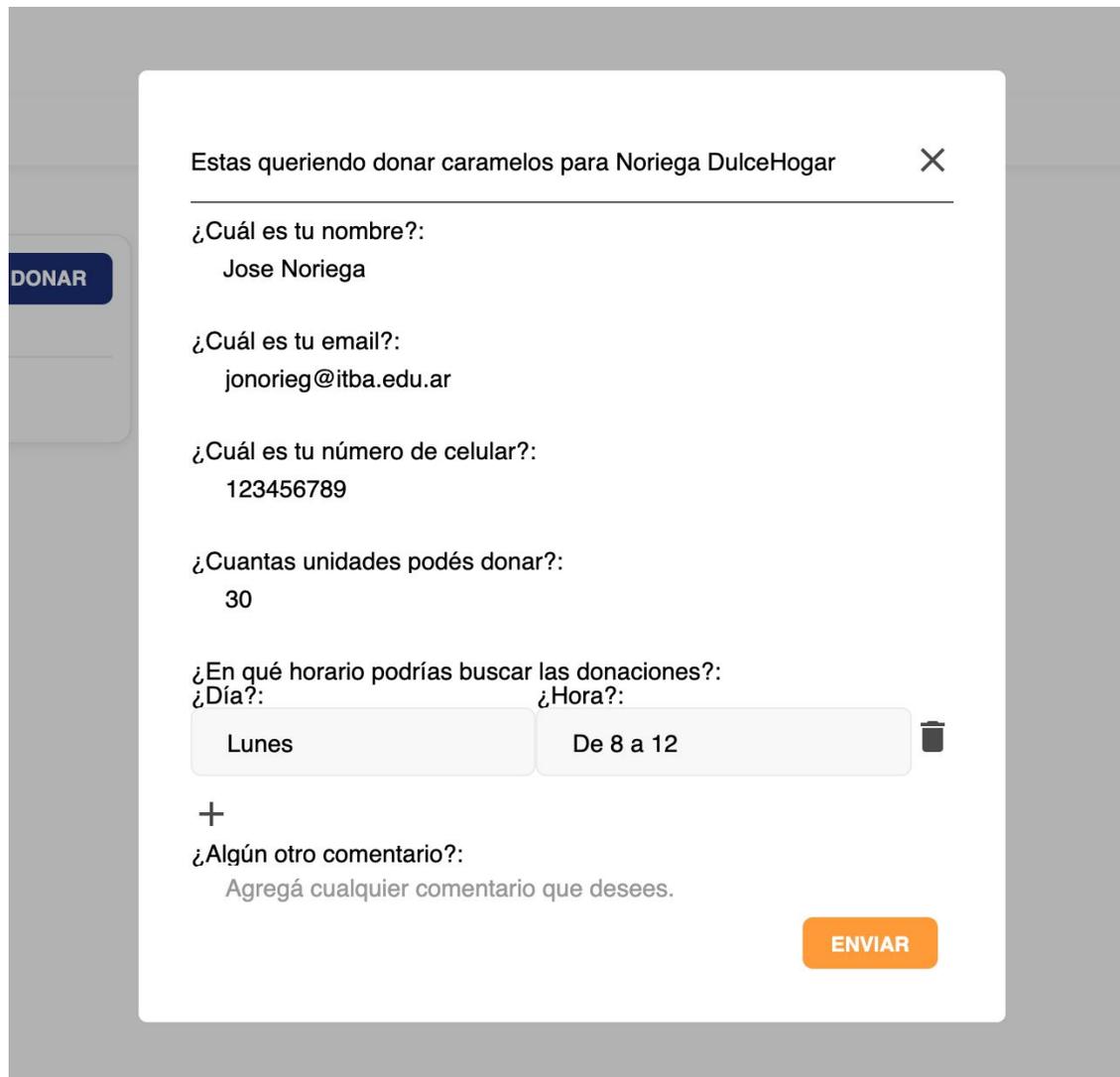
Noriega DulceHogar

DONAR

📍 Avenida Congreso 2077, 4B

Modal de donación

Permite donar a un pedido específico. Una vez creada dicha donación, el sistema busca un voluntario para poder transportarla.



Estas queriendo donar caramelos para Noriega DulceHogar ✕

¿Cuál es tu nombre?:
Jose Noriega

¿Cuál es tu email?:
jonorieg@itba.edu.ar

¿Cuál es tu número de celular?:
123456789

¿Cuántas unidades podés donar?:
30

¿En qué horario podrías buscar las donaciones?:
¿Día?: Lunes ¿Hora?: De 8 a 12 🗑️

+
¿Algún otro comentario?:
Agregá cualquier comentario que desees.

ENVIAR

Funcionalidades Web

La gran mayoría de las funcionalidades en Ponte Web, están relacionadas con el Front End, por lo tanto, la mayoría de las funcionalidades de backend no están implementadas por lo cual es necesario que las mismas sean llevadas a cabo mediante el dashboard de Forest.

Implementadas

- Registrarse/Iniciar sesión con un usuario. Este mismo puede arrancar siendo voluntario o seleccionar la opción más adelante
- Registrarse como Institución, pendiente a ser confirmado

- Publicar una donación o donar directamente a un pedido
- Cancelar publicación de donación.
- Buscar pedidos
- Actualizar el estado de una donación (Entregado, en camino, etc.)

No Implementadas

Las siguientes funcionalidades deben hacerse manualmente desde Forest

- Dar de alta una institución
- Publicar un pedido de una institución
- Vincular un pedido, una oferta y un voluntario
- Aprobación de solicitud para ser voluntario.

Funcionalidades a desarrollar

Los requisitos funcionales son muy similares a los requisitos de la versión Web, la versión de Android utilizara el mismo backend.

Autenticación

La aplicación debe permitir a un usuario crearse su propia cuenta, ingresando su nombre, apellido, dirección, email y una contraseña. El sistema le enviará un correo a la casilla de email proporcionada, para activar su nueva cuenta. El usuario creado por default siempre es un donante

Si el usuario desea involucrarse como voluntario, deberá hacer una solicitud mediante un formulario. La aplicación le requerirá al donante que ingrese: tamaño de donación que puede transportar (chica, mediana o grande), franjas horarias en las cuales está disponible para llevar la donación y algún comentario que el futuro voluntario desee aclarar. Una vez enviada dicha solicitud, el usuario pasará al estado de *quiero ser voluntario*, dónde deberá esperar a que la solicitud sea aprobada por algún administrador del sitio. Cuando sea aprobado, pasará a estado de *voluntario*, donde podrá ser seleccionado para trasladar donaciones.

Pedidos, ofertas y donaciones

La aplicación debe mostrar de forma sencilla todos los pedidos realizados por las instituciones, las ofertas hechas por el donante y, en caso de ser voluntario, las donaciones que le fueron asignadas.

En caso de ser un donante, el usuario deberá poder realizar una oferta, sin necesidad de un previo pedido o a un pedido explícito por parte de una institución. En dicha oferta deberá ingresar el bien de la donación, las unidades que desea donar, franjas horarias de disponibilidad y algún comentario explicativo en caso de ser necesario.

Cuando una oferta se corresponde con algún pedido realizado por alguna institución, en ese caso para denominarse donación. Dicha donación pasa por distintas etapas, desde su concepción hasta su entrega:

- Creada
- En búsqueda de institución
- En busca de un voluntario
- Voluntario asignado
- Siendo buscada por el voluntario
- Entregada por el donante
- Retirada por el voluntario
- Entregada
- Cancelada por el donante

Tanto el donante como el voluntario deberán poder interactuar con sus respectivas donaciones, llevando la misma hasta su estado final.

El donante podrá marcar una donación como entregada al voluntario o cancelarla en todo momento, excepto cuando ya entregó la misma.

El voluntario podrá marcar cuando está yendo a buscar la donación, cuando la retiró y cuando la entregó a la institución (destino). El voluntario no podrá cancelar la donación en ningún momento.

Instituciones

La aplicación le permitirá al usuario poder acceder a un listado de todas las instituciones inscritas y aprobadas en la plataforma.

Requisitos No Funcionales

A partir del relevamiento realizado sobre Ponte Web y los requisitos funcionales definidos previamente, se decidieron los siguientes atributos de calidad

Confiabilidad

La aplicación debe ser tolerante a fallos, es decir debe seguir en funcionamiento a pesar de alguna falla de alguno o varios de sus componentes. Debe informar en el caso de errores que sean relevantes al usuario, sin la necesidad de ser demasiado explicativos ya que el usuario carece de perfil técnico. Si debe proveer soluciones o alternativas en caso de ser necesario.

Simplicidad

Simplicidad es ausencia de complejidad o dificultades. Si bien Ponte está siendo rediseñado en cuanto a su arquitectura en el Proyecto Final Ponte 4, se encontraron ciertas limitaciones en el flujo por lo que la simplicidad debe ser un atributo importante a lograr para que la aplicación erradique la mayor cantidad de complejidad posible al usuario para que logre su objetivo, sin necesidad de un tercero o un administrador.

Flexibilidad

También llamado modificabilidad, es la capacidad del sistema de admitir cambios, ya sea por algún error o cambio de requerimientos. El diseño arquitectónico y de interfaz de la aplicación, debe estar instrumentado desde esta perspectiva, separando y modularizando la aplicación en componentes reutilizables. A su vez, debe permitirle al usuario poder corregir o volver atrás en algún error que haya cometido

Usabilidad

Relacionado a Simplicidad, el diseño UI debe ser lo más sencillo y explicativo a la vez, para facilitar la curva de aprendizaje del usuario. El usuario no debería requerir ningún tutorial o explicativo para usar la aplicación.

Performance

La aplicación debe ser eficiente en el uso de recursos del sistema en materia de memoria, procesamiento y uso de redes. Debido a que la aplicación no permitirá por el momento el modo offline, se debe optimizar el uso de llamadas al API del backend.

Seguridad

La aplicación comprobará el acceso a la misma mediante la identificación del usuario. No se podrá utilizar la aplicación y no se podrán acceder a los datos de la misma si el usuario no está logueado o registrado. A su vez, hay muchas mejoras para realizar que están detalladas más adelante en este informe

Extensibilidad

El diseño y la arquitectura deben estar pensados para que la aplicación sea fácilmente extensible. Agregar nuevas features a la misma no debería requerir grandes cambios y tampoco comprometer lo que ya está implementado.

Desarrollo y QA

El desarrollo contempla todas las funcionalidades mencionadas en el punto anterior. Se llevará a cabo en sprints que durarán entre tres semanas y un mes, enviando versiones para recibir feedback de parte de la tutora. Se utilizó Trello para organizar las tareas de cada sprint (Ver Figura 1).

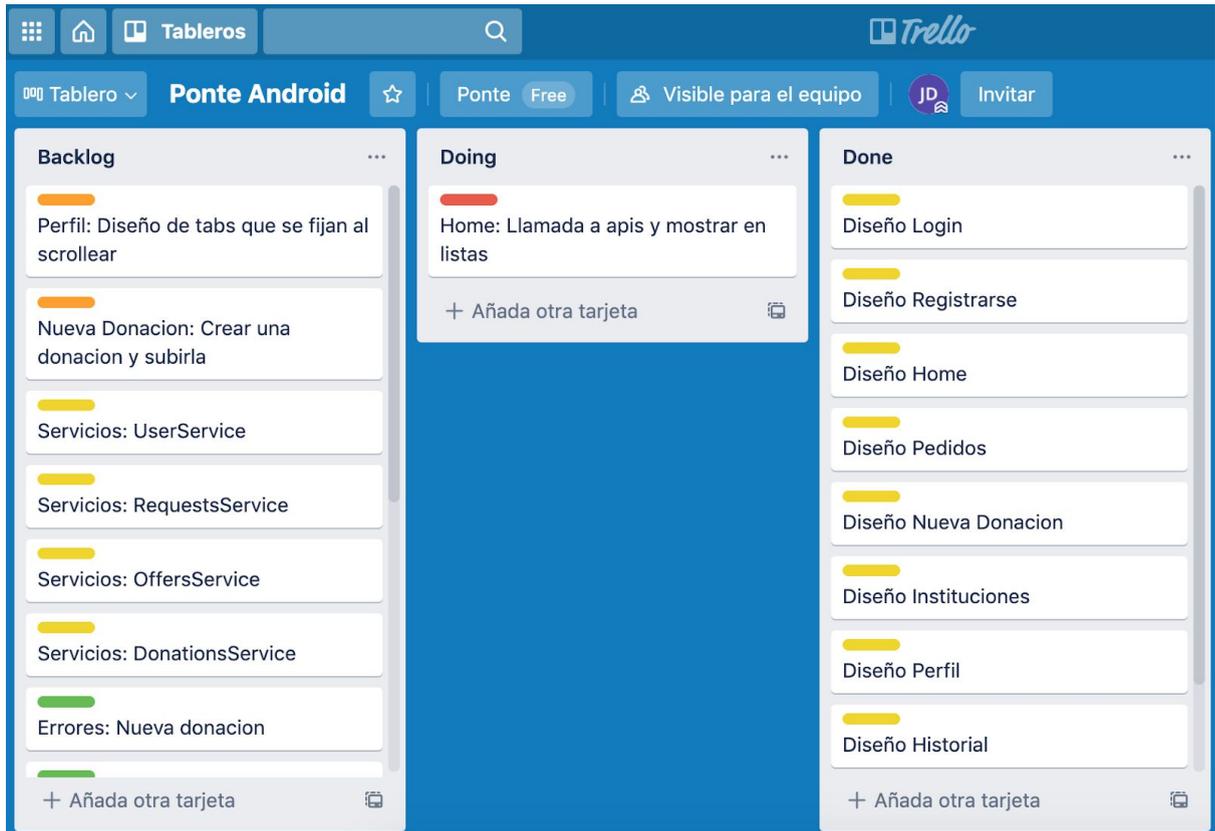


Figura 1: Tablero de Trello para organizar las tareas durante el desarrollo.

Diseño de Flujo y UI

El desafío para la versión Mobile consiste en aprovechar el limitado espacio que se tiene en la pantalla y a la vez mantener una buena estética. Las pantallas no deben abrumar al usuario con información que no necesite manteniendo el atributo de *simplicidad* y su uso debe ser lo más intuitivo posible; el usuario no debería necesitar un instructivo para moverse y manejar dentro de la aplicación.

Desde el punto de vista estético, es recomendable seguir los lineamientos de *Material Design*, en el uso de iconografía, componentes y gama de colores.

Para encarar este desafío, se utilizó Sketch, un programa de diseño profesional, que permite integrar los componentes de Material Design, y también permite armar un “preview” de tu aplicación, que permite probar el flujo de la misma.

A su vez, se hizo un relevamiento exhaustivo de aplicaciones existentes, para tomar las mejores prácticas de las mismas. Al ser aplicaciones de uso masivo, permite aplanar la curva de aprendizaje de nuestra aplicación y de esta forma cumplir el atributo de *usabilidad* mencionado anteriormente.

En base a lo ya existente, se hicieron diseños que sigan con la estética de Ponte y un flujo adaptado para Mobile.

Paleta de colores

Se siguió la línea estética de lo que ya está desarrollado en Ponte Web. Simplemente se adaptaron los colores a los Recomendados por Material Design. Ver Figura 2

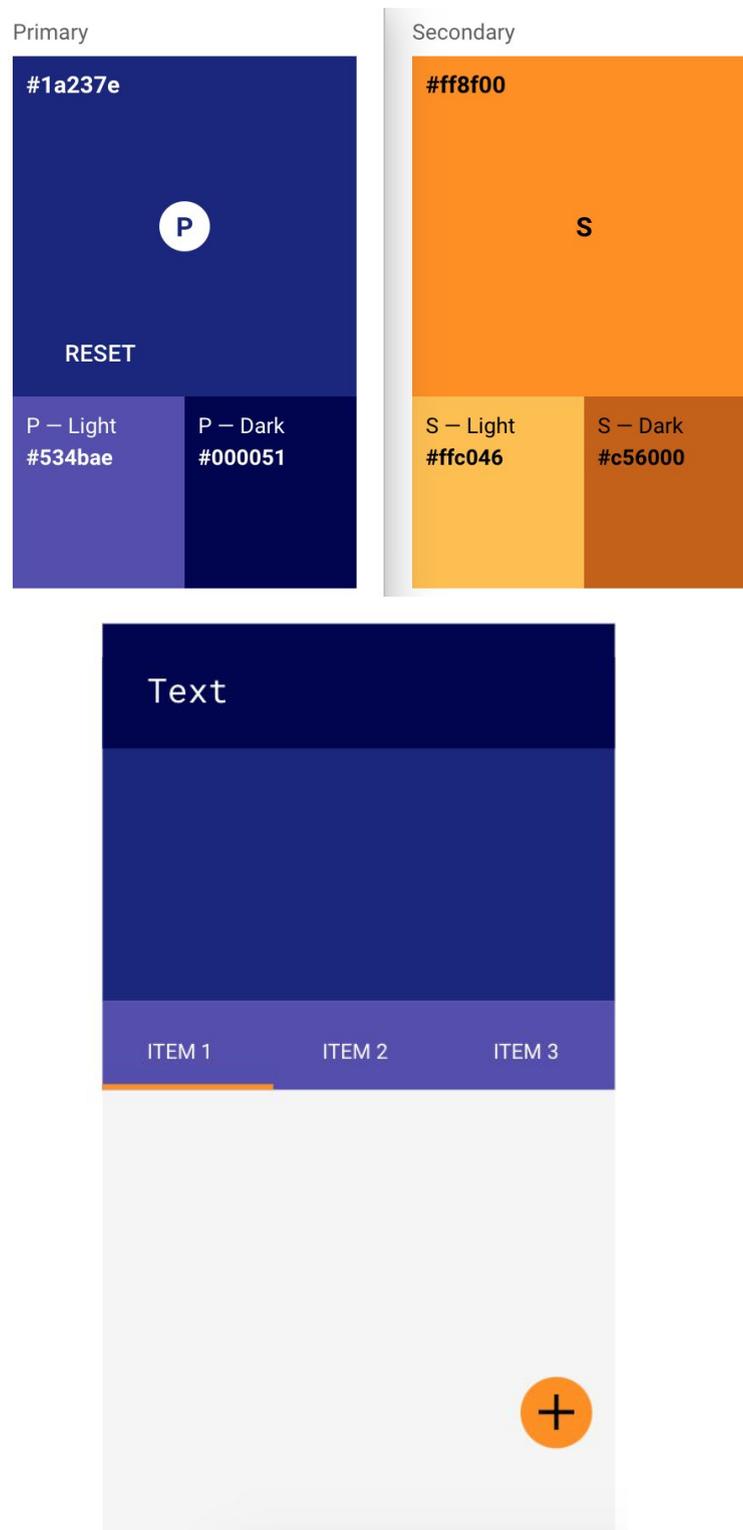


Figura 2: paleta de colores adapta a los lineamientos de Material Design

Navegación

Se plantearon dos modelos de navegación para la aplicación; el navigation drawer y el bottom navigation, cómo se ven en la *Figura 3*. La ventaja del navigation drawer, es que es fácil de escalar, ya que es cuestión de agregar más secciones al drawer, pero requiere de un tap extra para abrir el drawer. En cambio, el bottom navigation permite una navegación más accesible ya que está más cerca del pulgar (*Figura 4*) y con un solo tap, pero solo se permite entre 3 y 5 accesos. Se optó por utilizar el bottom navigation, ya que nuestra aplicación no necesita por el momento más accesos, además de ser la práctica más utilizada por varias aplicaciones (*Figura 5*).

Eventualmente, se podrían mezclar ambos tipos de navegación como lo hace *Google Maps* (*Figura 6*)

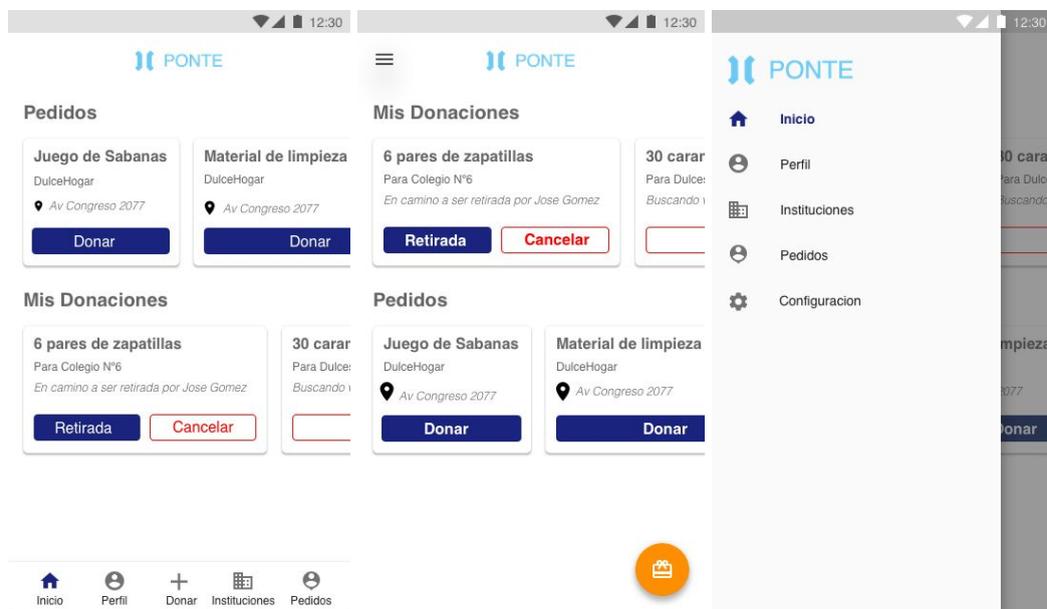


Figura 3: El bottom navigation se ve en la imagen de más a la izquierda vs el navigation drawer abierto y cerrado.

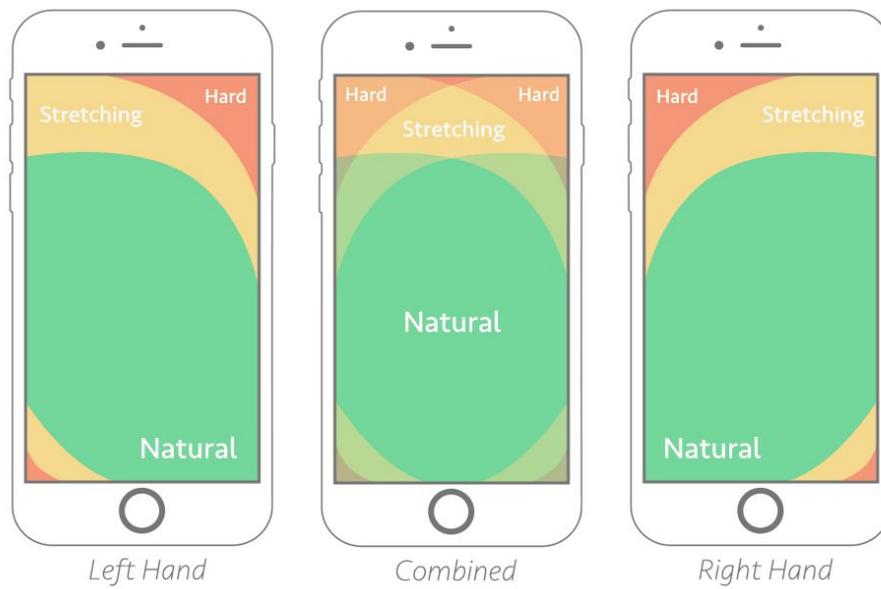


Figura 4: mapa que muestra con qué dificultad se accede a las distintas zonas de la pantalla. El bottom navigation tiene mejor acceso que el navigation drawer.

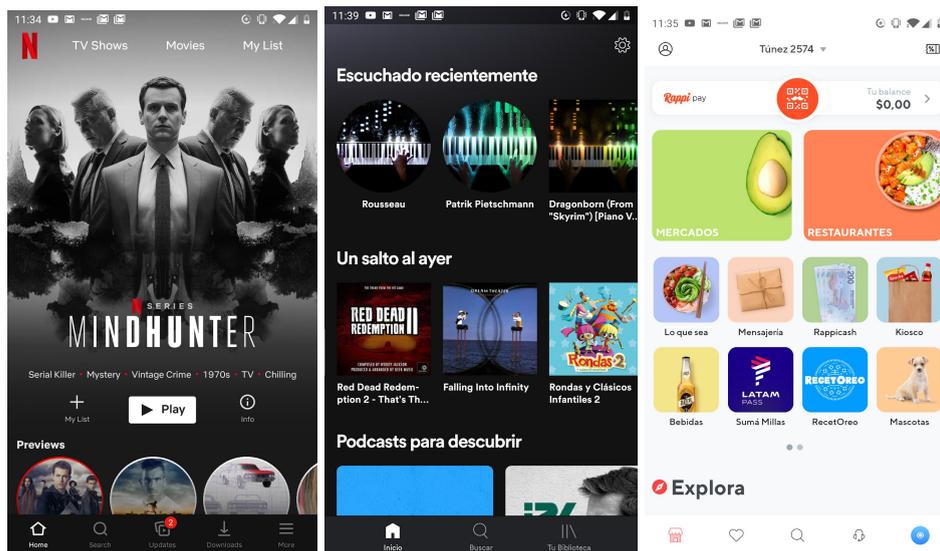


Figura 5: Netflix, Spotify y Rappi con bottom navigation

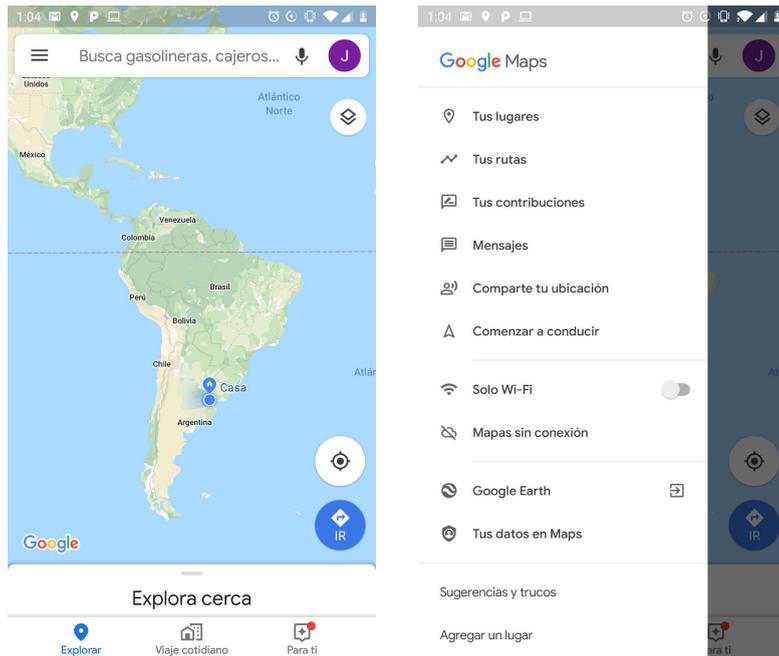


Figura 6: Google Maps tiene una combinación entre navigation drawer y el bottom navigation

Home

El diseño de la home intenta encapsular la mayor cantidad de acciones que puede realizar el usuario. Se optó por un diseño tipo “cards” con scroll horizontal al estilo Rappi, en donde se muestran los Pedidos primero y luego Mis Donaciones si el usuario es un donante, y se le agrega la sección de Mis Entregas en primer lugar si el usuario es Voluntario. Las ventajas de este diseño es que permite al usuario realizar las acciones principales de la plataforma en la primera pantalla: marcar el estado de una donación, donar a un pedido y actualizar una entrega, en el caso de ser un voluntario. También es escalable, ya que fácilmente permite agregar más secciones.

Para hacer una nueva donación, primero se incluyó un botón flotante sobre las secciones de la Home, pero terminó resultando medio confuso ya que podía tapar otras acciones del usuario. Se optó por incluir la acción de la nueva donación en el bottom navigation, ya que permite que se pueda usar desde cualquier sección de la app y no estar limitada a una pantalla. Ver figuras 7 y 8.

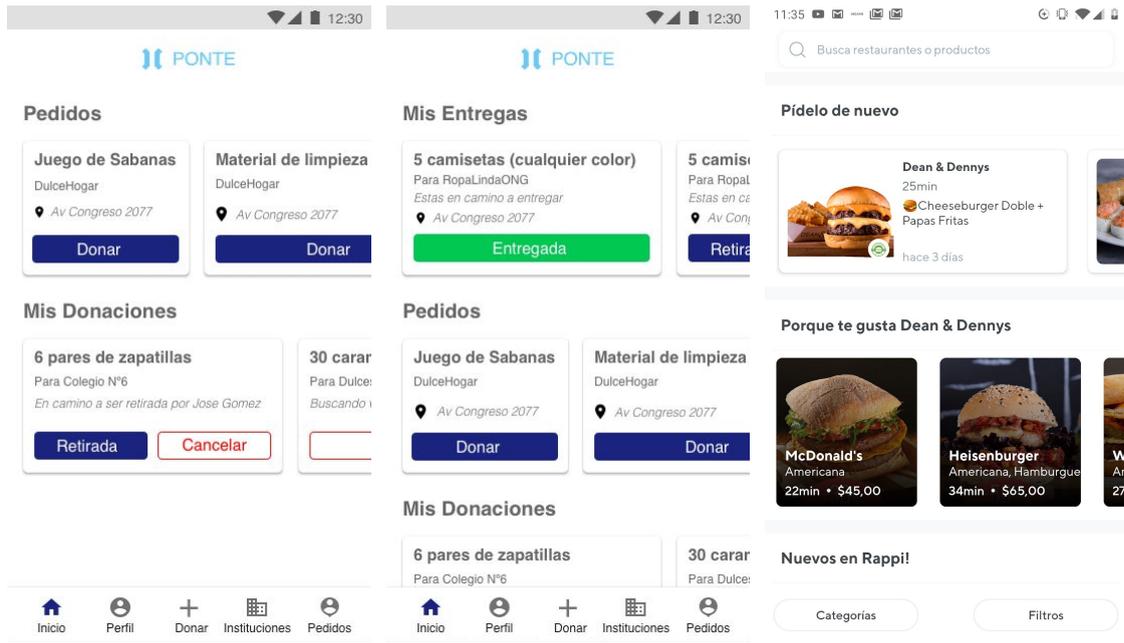


Figura 7: pantalla de la Home con el diseño de cards horizontal, similar al que tiene Rappi.

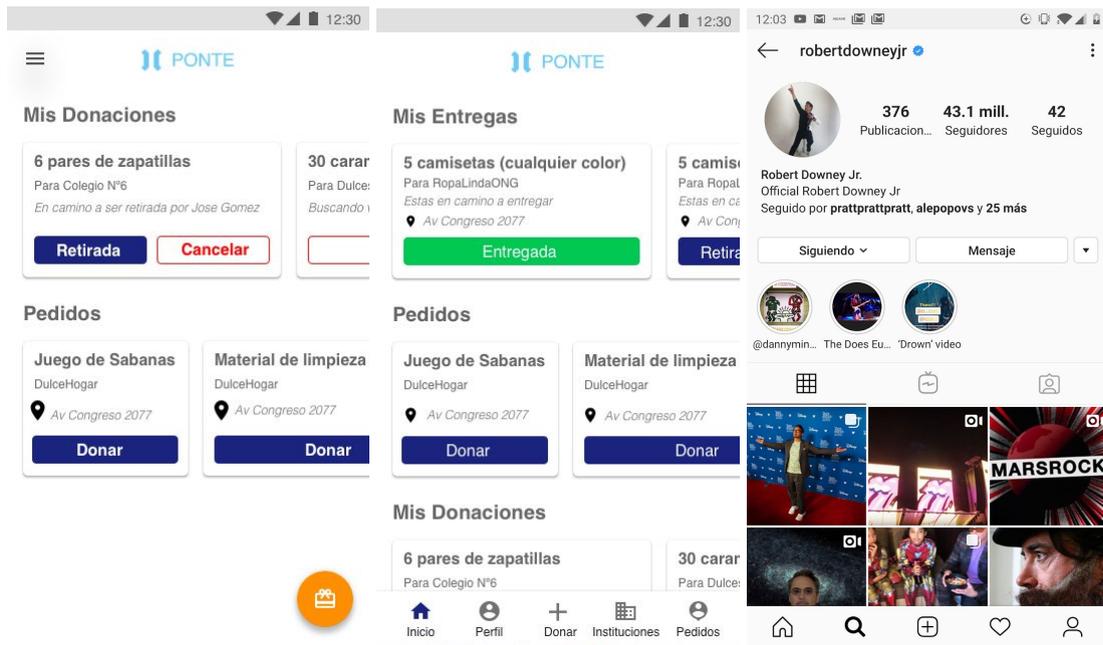


Figura 8: primera propuesta con botón flotante para una nueva donación, luego la propuesta final con la acción en el bottom navigation, cómo tiene Instagram.

Donar

Como se mencionó anteriormente, una nueva donación se puede hacer mediante el bottom navigation, ya que garantiza accesibilidad en casi toda la aplicación. Al seleccionar, aparece desde abajo la pantalla para hacer la donación, tapando el bottom navigation. Este diseño va a en contra de los lineamientos de Android, ya que establece que ninguna sección del bottom navigation debe tapar la navegación, pero encontramos que es la mejor manera para que el usuario pueda realizar esta acción en cualquier momento, y el trade-off de tapar la navegación es virtualmente inexistente.

El diseño le permite al usuario ingresar los campos necesarios para realizar una donación. Para las franjas horarias, se siguieron los lineamientos del diseño original de Ponte, aunque a futuro se podría usar algún componente nativo para seleccionar franjas horarias. El botón de enviar este fijado en la parte inferior de la pantalla, para facilitar el acceso y siempre tener visibilidad

Si el usuario desea donar un pedido, el diseño de la pantalla es muy similar, solo que el objeto a donar no lo puede cambiar. *Ver Figura 9*

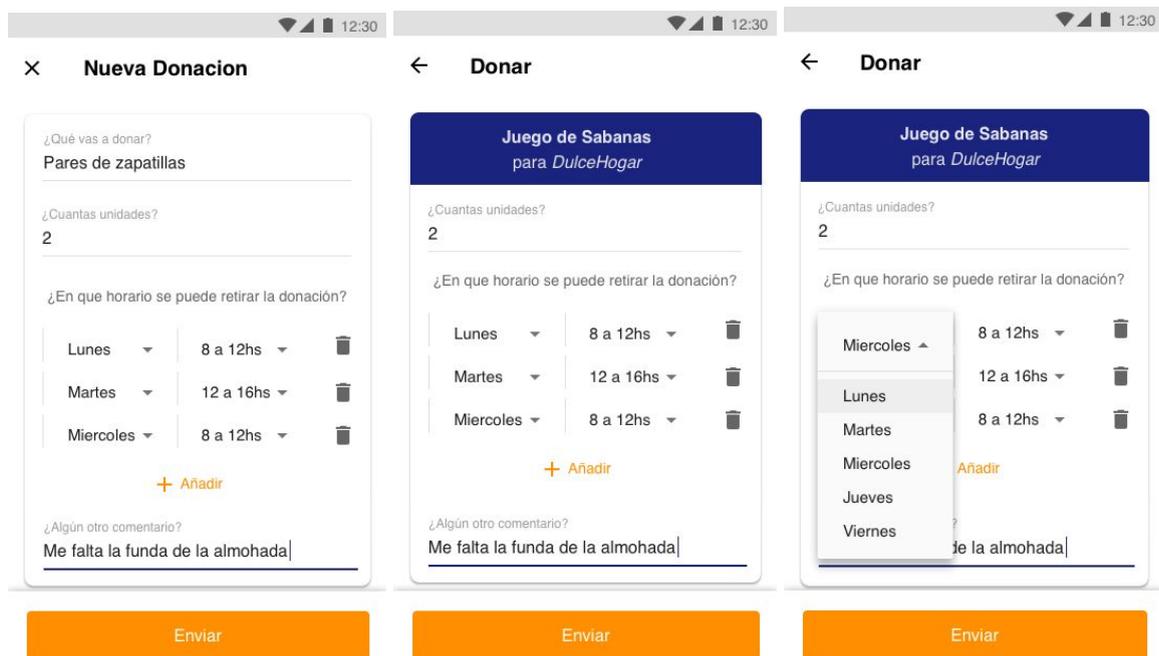


Figura 9: Nueva donación y donar un pedido existente.

Perfil

En el perfil el usuario va a poder visualizar sus datos, elegir si desea ser voluntario y visualizar sus donaciones y entregas (en caso de ser un voluntario). Para los listados se optó por utilizar tabs (la tab solo se mostraría si el usuario es voluntario) y para aprovechar el espacio, cuando el usuario scrolle por cualquiera de las listas, las tabs terminan ocupando toda la pantalla. En este caso, tomamos como modelo a Instagram, ya que se maneja muy bien la visualización de mucha información de manera intuitiva. Ver Figura 10

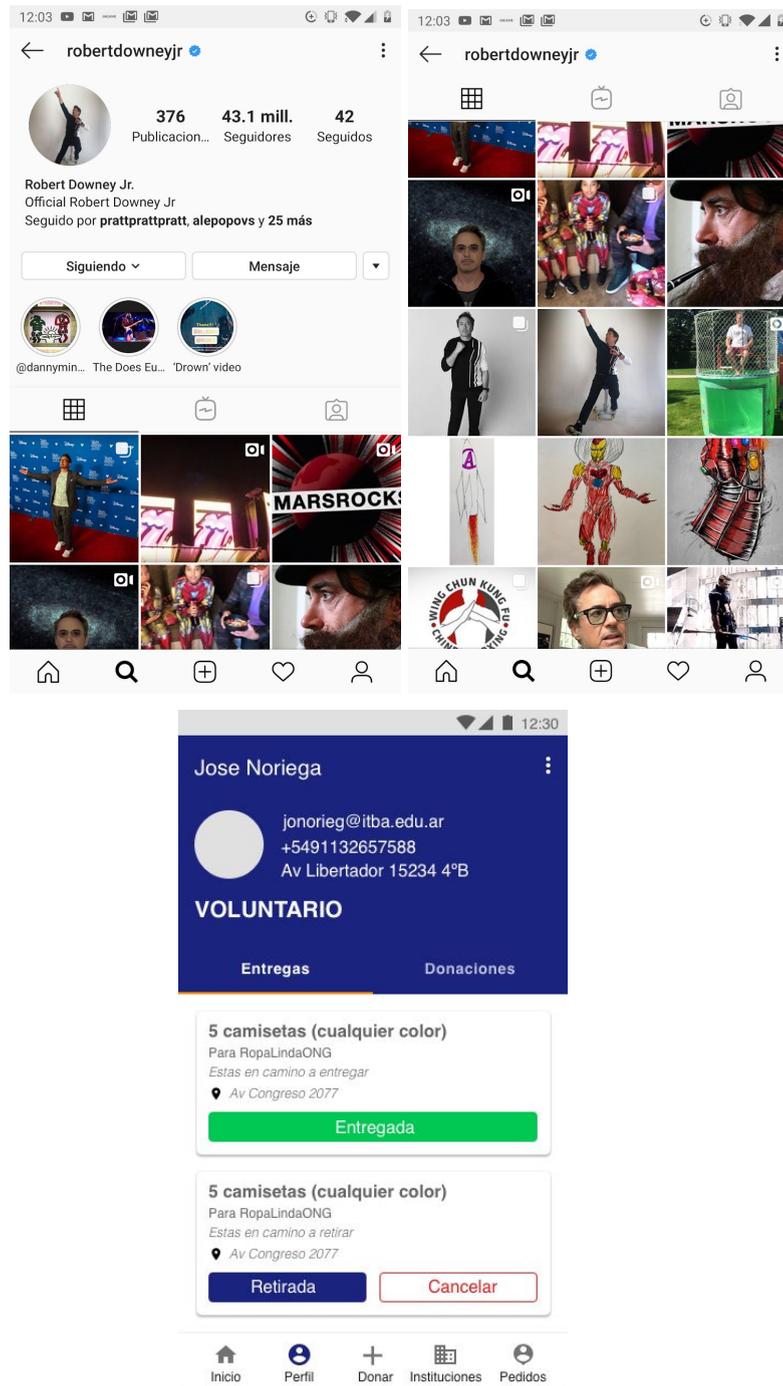


Figura 10: el perfil permite scrollear por las fotos y las tabs quedan fijas en la parte superior de la pantalla

También se decidió por poner el acceso a configuración en el perfil. Originalmente iba a estar en el bottom navigation, pero termina ocupando un acceso importante cuando la idea es que no se acceda a configuración de forma recurrente. Por el momento la configuración consiste de un acceso al historial de donaciones y entregas, información de la aplicación y el logout. No es deseable que el usuario se deslogue de la aplicación, entonces se suele colocar en configuración/ajustes. Ver figura 11

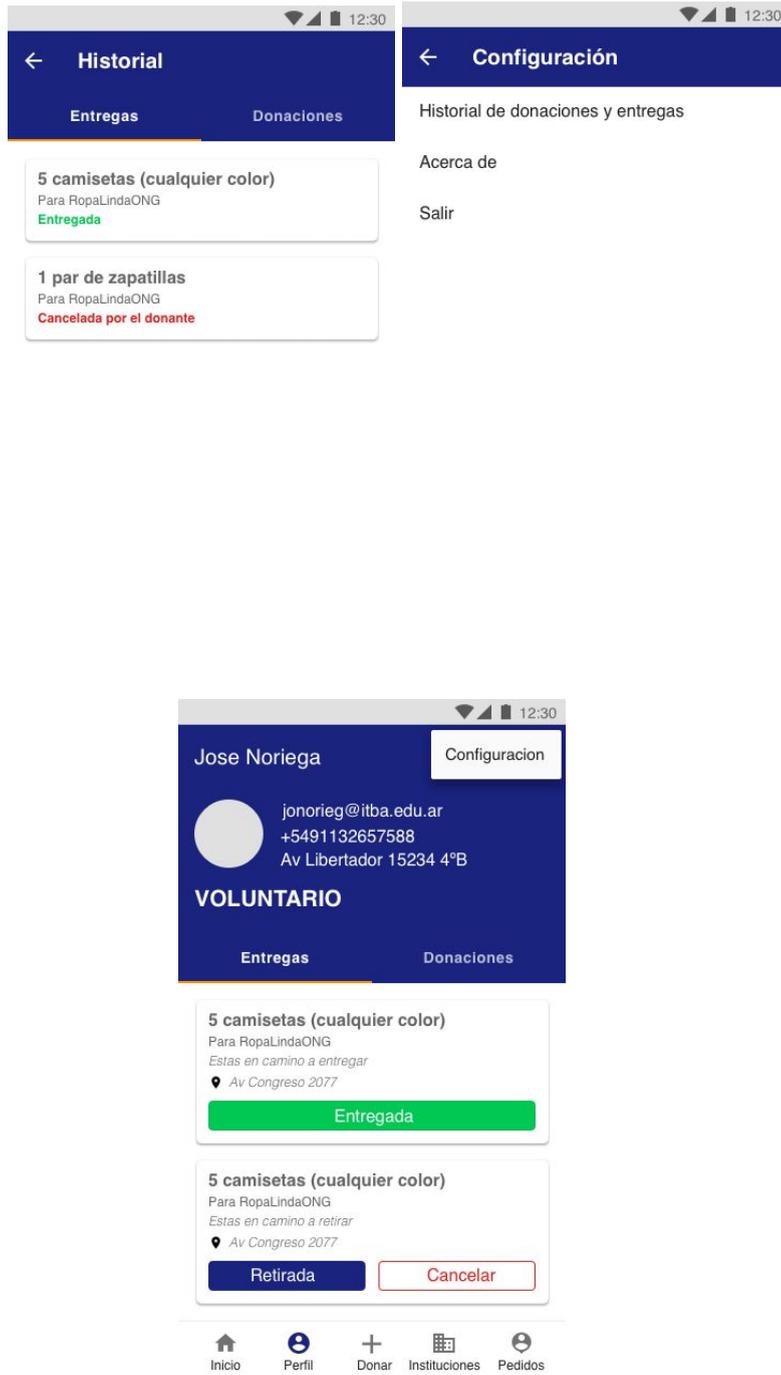


Figura 11: el acceso a configuración desde el perfil, las configuraciones y el listado del historial

Instituciones y pedidos

Las secciones de instituciones y pedidos son muy sencillas, consisten de un listado de “cards” de instituciones y pedidos. Ver figura 12

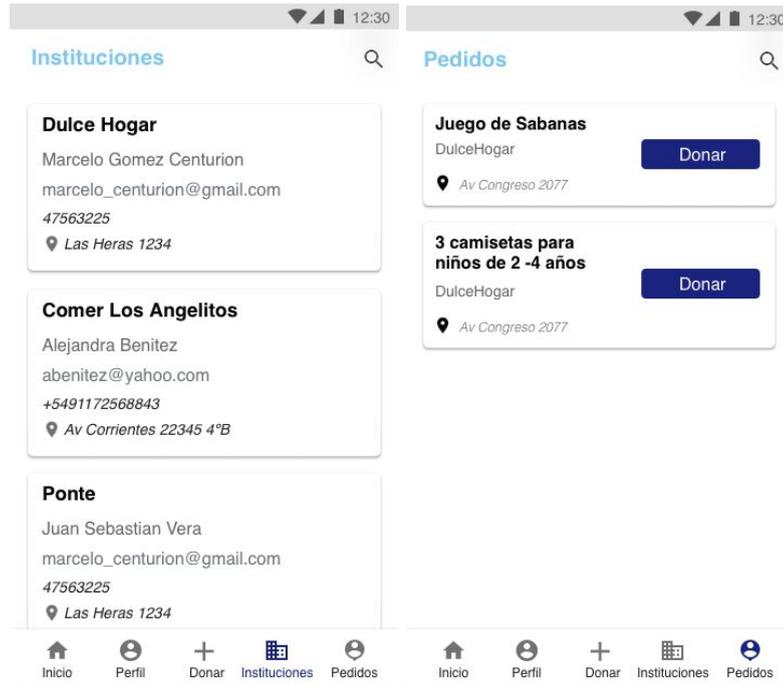


Figura 12: listado de instituciones y de pedidos

Pantalla Login/Registrarme

La pantalla de login es muy sencilla, ya que consiste del título y logo de la aplicación, dos entradas de texto para el email y la contraseña, un botón para entrar y otro para registrarse.

El registro consiste en un formulario en donde se llenan todos los datos del usuario. El diseño final se puede visualizar en la Figura 17 y Figura 18

Arquitectura

La arquitectura de cualquier plataforma o aplicación es el aspecto más importante del desarrollo, ya que sienta los cimientos por sobre los cuales se va a construir nuestro software. Una buena arquitectura nos permite tener código limpio, modularizado y a su vez, permite que cualquier desarrollador pueda entenderlo y expandirlo. Es importante tomarse el tiempo para decidir qué arquitectura vamos a desarrollar, teniendo en cuenta su robustez, escalabilidad y elasticidad.

El principio más importante a seguir es la separación de problemas, que consiste en dividir la lógica de UI, de negocios y del modelo o persistencia. Separando la lógica, nos permite reutilizar código; la lógica de negocios puede ser compartida entre varias vistas. También permite un fácil mantenimiento, ya que cuando aparece algún bug, podemos encontrarlo más rápido en el código.

El patrón de diseño que vamos a utilizar para nuestra arquitectura es **Model-View-ViewModel (MVVM)**. Este patrón, como su nombre lo indica, tiene tres componentes: la vista (view), el viewmodel y el modelo.

La vista, contiene toda la lógica relacionada a la UI (colores, animaciones, fuentes, etc.) y, en el caso Android, las interacciones con el sistema operativo.

El ViewModel contiene la lógica de presentación y puede contener lógica de negocio, dependiendo de la naturaleza de tu aplicación.

El modelo, también puede tener lógica de negocio, pero es principalmente encargada de la persistencia, ya sea persistencia externa a una api o a una base de datos local.

Los componentes interactúan entre sí mediante eventos, y cada componente suscribe u observa los eventos que le corresponden de los otros componentes. Ver figura 13

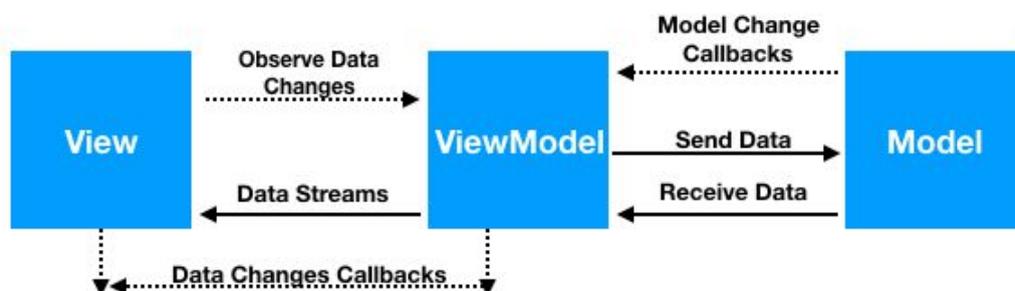


Figura 13: MVVM, como se relacionan los tres componentes¹

¹ Referencia extraída del sitio www.journaldev.com

Desarrollo

La aplicación se podría haber desarrollado tanto en *Java* como en *Kotlin*, ya que Android soporta ambos lenguajes y pueden convivir ambos lenguajes en la misma aplicación. Se optó por *Kotlin* ya que permite escribir lo mismo que *Java* pero con mucha menor cantidad de código, además de ser el lenguaje que Android optara para el futuro. El *IDE* utilizado es Android Studio.

Para nuestra arquitectura, disponemos del pack de librerías de Android llamado Jetpack que está compuesta en 4 patas: Architecture, UI, Foundation y Behavior. El único conjunto que no fue necesario en esta etapa de Ponte es Behavior. Ver Figura 14



Figura 14: las librerías de Android JetPack ²

² Referencia extraída del sitio <https://medium.com/orbismobile/qu%C3%A9-es-y-qu%C3%A9-no-es-android-jetpack-1248d1f36670>

Architecture

El conjunto de librerías de arquitectura es la más importante de todas, de las cuales se utilizaron principalmente las librerías de *Viewmodel*, *DataBinding*, *Navigation* y *Lifecycles* entre otros.

La mayor complejidad del desarrollo en Android, es el manejo del ciclo de vida de las Activities, las cuales son el contrato entre el sistema operativo (SO) y nuestra aplicación. En ellas se incluye toda la lógica que corresponde al UI, osea la capa View en nuestro MVVM, y las llamadas del SO que nos informara el estado (la “vida”) de nuestra Activity cuando se pausa, se destruye, se reanuda, etc etc. Esto puede ocasionar fallas en nuestra aplicación, ya que se podría tener una referencia a una vista en la capa de presentación, la cual ya había sido destruida por el SO (aplicación en background, falta de memoria, etc), llevando crashes. Son errores muy difíciles de reproducir y de identificar al momento del desarrollo. Es por eso que se utiliza la librería Viewmodel.

Viewmodel nos permite crear viewmodels “out of the box”, es decir, extender la clase ViewModel nativa para poder escribir nuestras propias implementaciones con la lógica de negocios. Es aquí donde entra en juego también la librería de *lifecycle*, en la forma de objetos de tipo ObservableField y MutableLiveData, en donde insertaremos nuestros flujos de datos, y serán observados por los subscriptores que los necesiten.

Los objetos de tipo ObservableField representan los distintos estados de la vista: visibilidad, textos, valores numéricos, etc, siendo siempre valores built-in (int, boolean, string, etc) los cuales van a estar referenciados en el archivo XML de la vista.

Los MutableLiveData son para datos más complejos, como listas de donaciones, pedidos, instituciones etc. los cuales pueden ser observados por “observer” (valga la redundancia). En cierto sentido se podría hacer esta misma implementación utilizando Rx para Kotlin, pero la ventaja de MutableLiveData, es que para ser observado se le debe “pasar por parámetro” un objeto de tipo Lifecycle, que suele ser una Activity o Fragment de Android. En resumidas cuentas, MutableLiveData conoce los ciclos de vida de “observadores” que estén suscritos, solo pasando datos a los que estén en un ciclo de vida activo.

Nuestras Activities o Fragments comparten Viewmodels. ¿Cómo se logra esto sin tener que pasar nuestro Viewmodel vía constructor o setters? ¿Cómo nos aseguramos de que ninguna instancia de un Viewmodel quede ejecutando en “background”? Es aquí donde utilizamos la librería de *Databinding*.

Databinding nos permite unir XML, Activity/Fragment con nuestro ViewModel. Mediante el uso de databindings, podemos hacer referencia a variables de instancia del ViewModel en nuestro XML. También le permite nuestra Activity o Fragment obtener un ViewModel en particular, sin crear una nueva instancia, ya que mantiene un registro interno de todos los diferentes ViewModels instanciados. Si nuestro tipo de ViewModel no fue instanciado aún, nos devolverá una instancia nueva. Ver Figura 15

En Ponte Android, tenemos *HomeViewModel*, que utiliza todo lo mencionado anteriormente, para ser compartido entre las principales secciones de la aplicación.

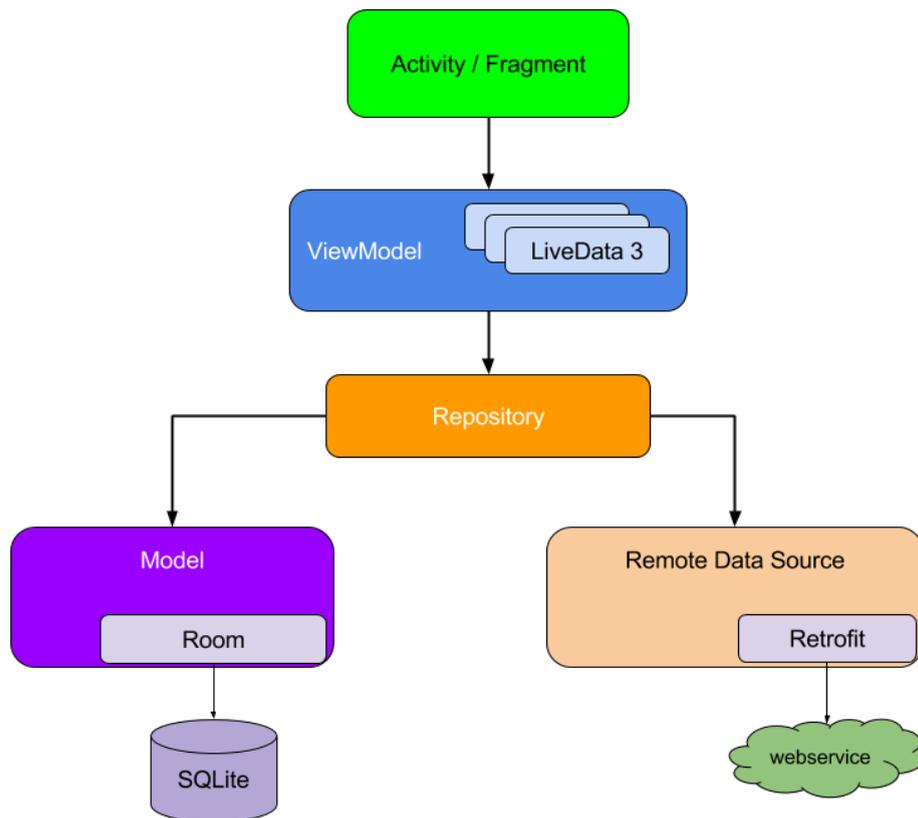


Figura 15: arquitectura MVVM con componentes de Android (ref: developer.android.com)³

Navigation nos proporciona una implementación sencilla del BottomNavigation, mencionado anteriormente en la sección de Diseño y Flujo UI, que a su vez está embebido con el Android Studio, lo cual nos permite ver un grafo de nuestra navegación, así como agregar transiciones, animaciones, etc. Ver Figura 16

³ Referencia extraída del sitio developer.android.com

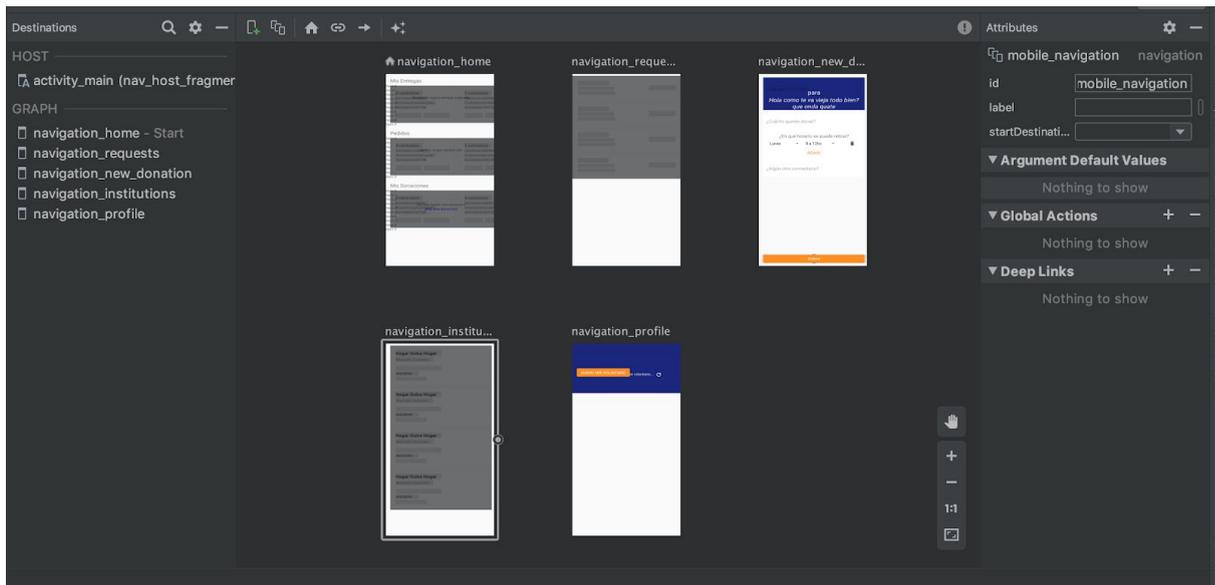


Figura 16: grafo de navegación desde el Android Studio, embebido con JetPack

Repository

El repositorio es la capa que se encarga de la obtención y persistencia de los modelos. Estos pueden ser persistentes remota o localmente. La aplicación no va a soportar modo *offline*, así que los datos se mantendrán remotamente a excepción de las credenciales del usuario, que se guardan localmente en el SharedPreference. Para esto último, se incorporó Hawk, una librería que encapsula al SharedPreference nativo, y permite un uso directo y simplificado del mismo

Para usar la API de Ponte, se utilizó la librería *Retrofit* y *OkHttp*, creadas por *Square*. Son las librerías más utilizadas para comunicaciones con APIs en general, con una implementación sencilla e intuitiva. Las llamadas a la API se encapsulan dentro de Corrutinas (Coroutines) de Kotlin, lo cual permite que llamadas bloqueantes a IO, se ejecuten de forma concurrente, sin la necesidad de crear nuevos Threads cada vez.

Foundation y UI

Foundation se utilizó para el soporte MultiDex de la aplicación, que amplía la cantidad de métodos que se pueden utilizar en la misma. También se utilizó Android KTX, que contiene varias de las nuevas librerías de AndroidX, que terminará reemplazando a AppCompat (librería que permite compatibilidad hacia versiones previas de Android). Pero principalmente permite extensiones para un Kotlin idiomático para Android que nos provee de:

- Extensión de propiedades y funciones, podemos agregar funciones a clases ya existentes sin la necesidad de extender la clase en sí
 - Parámetros nombrables: se puede nombrar qué argumentos se le desean pasar valores por parámetro.
 - Corrutinas, mencionado anteriormente

UI es el conjunto de librerías clásicas de Android que contiene Activity y Fragments, pero a su vez los distintos tipos de Layouts que ofrece Android.

Aplicación

A continuación se mostrarán las imágenes finales de la aplicación, separadas por su correspondientes secciones. Las mismas fueron tomadas desde el emulador de Android Studio

Login

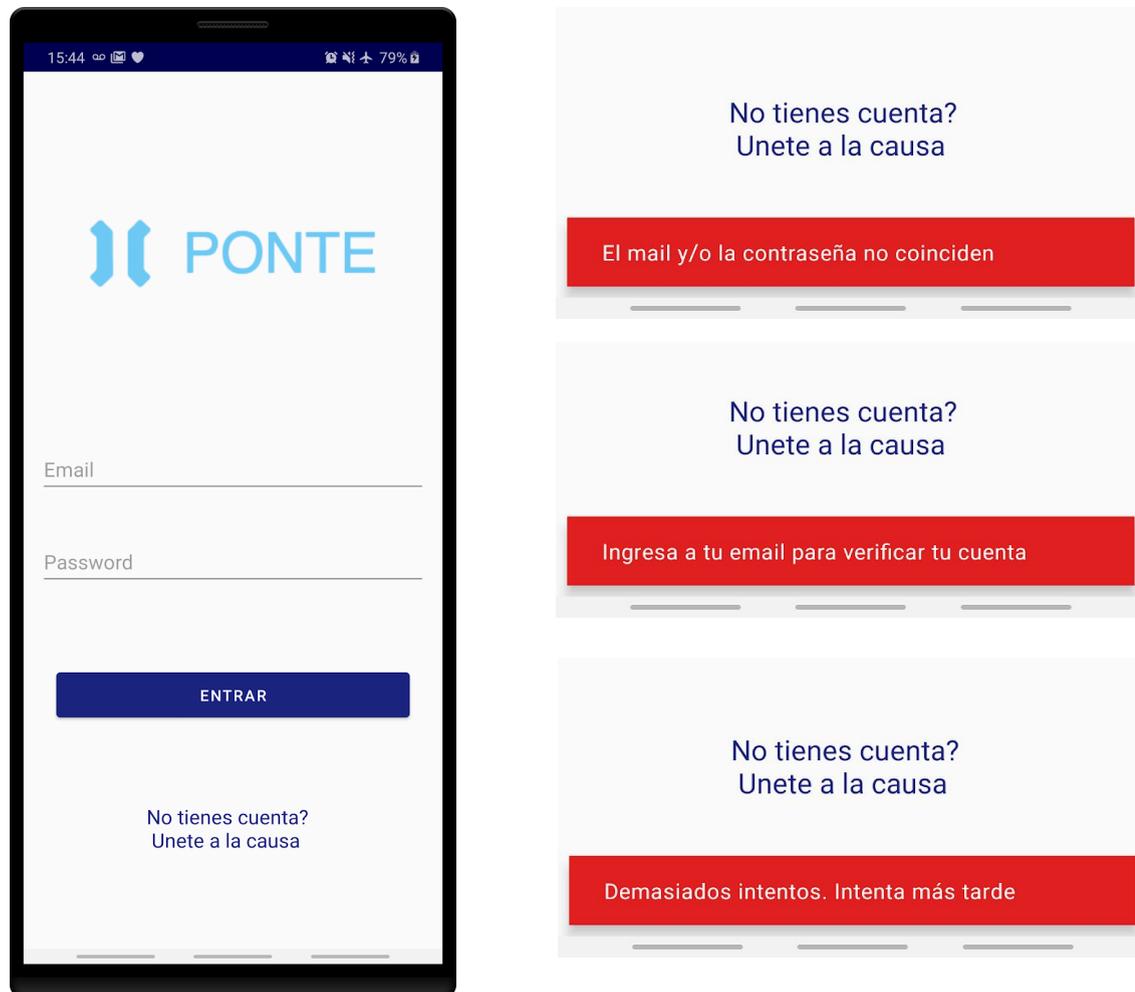


Figura 17: pantalla de login con sus tres tipos de errores

El login requirió un diseño sencillo, donde el usuario solo ingrese su mail y contraseña. También posee un enlace que lo lleva a registrarse.

Se consideraron tres tipos de error: mail y/o contraseña incorrectos, mail no verificado y demasiados intentos con un mail en particular.

Registrarse

La pantalla de registrarse consiste en un formulario con 6 campos: nombre, apellido, dirección, teléfono, mail y contraseña (como se puede ver en la Figura 18). A su vez, presenta varios tipos de errores (Figura 19) que incluyen: campos vacíos, error genérico, mail en uso y contraseña muy corta.

Una vez que el usuario se haya registrado con éxito, se le mostrará una confirmación del mismo para que verifique su email (Figura 20)

The image displays two mobile phone screens showing a registration form titled "Regístrate".

The left screen shows the form with empty input fields for:

- Nombre
- Apellido
- Dirección
- Teléfono
- Email
- Contraseña (with a toggle icon for visibility)

The right screen shows the form filled with the following data:

- Nombre: Jose
- Apellido: Noriega
- Dirección: Calle 1234
- Teléfono: 123567890
- Email: test@gmail.com
- Contraseña: masked with dots (with a toggle icon for visibility)

Both screens feature a blue button at the bottom labeled "REGISTRARME".

Figura 18: formulario para registrarse a Ponte

15:47 79%

Regístrate

Nombre

Tienes que poner tu nombre

Apellido

Tienes que poner tu apellido

Dirección

Tienes que poner tu dirección

Teléfono

Tienes que poner tu teléfono

Email

Tienes que poner tu email

Contraseña

Tienes que elegir una contraseña

REGISTRARME

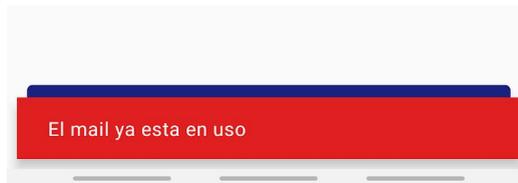
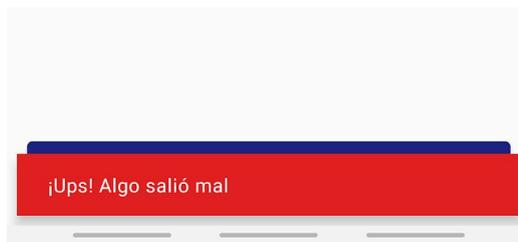


Figura 19: los errores que se pueden presentar al registrarse

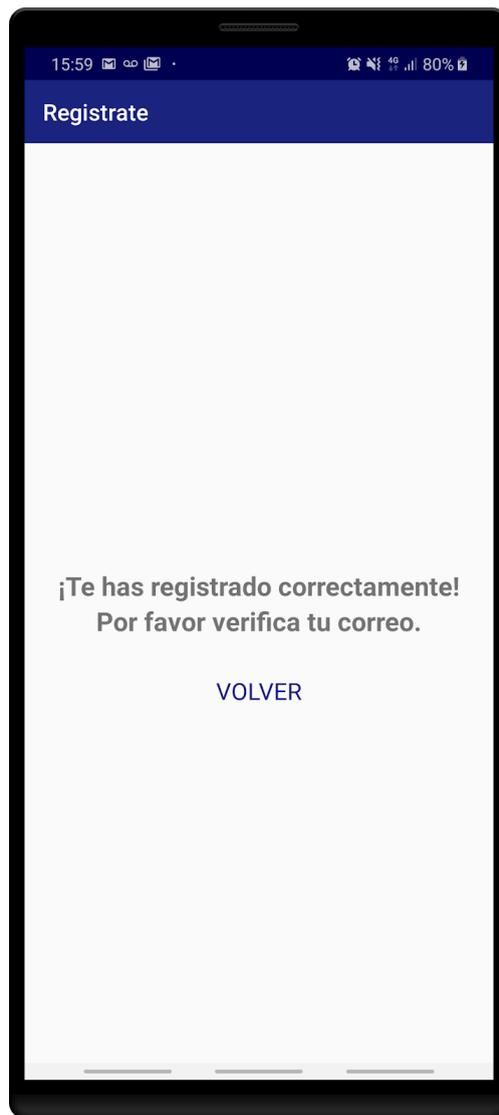


Figura 20: confirmación de que se ha registrado con éxito.

Home

La Home se podría considerar la sección más importante de la aplicación, ya que es un pequeño dashboard que nos muestra el estado en el que se encuentran las entregas, donaciones y pedidos. También podemos interactuar con cada uno de ellos.

La figura 21 muestra cómo sería el típico home de un donante, en cambio, la figura 22 muestra el dashboard de un voluntario, son sus entregas en la parte superior.

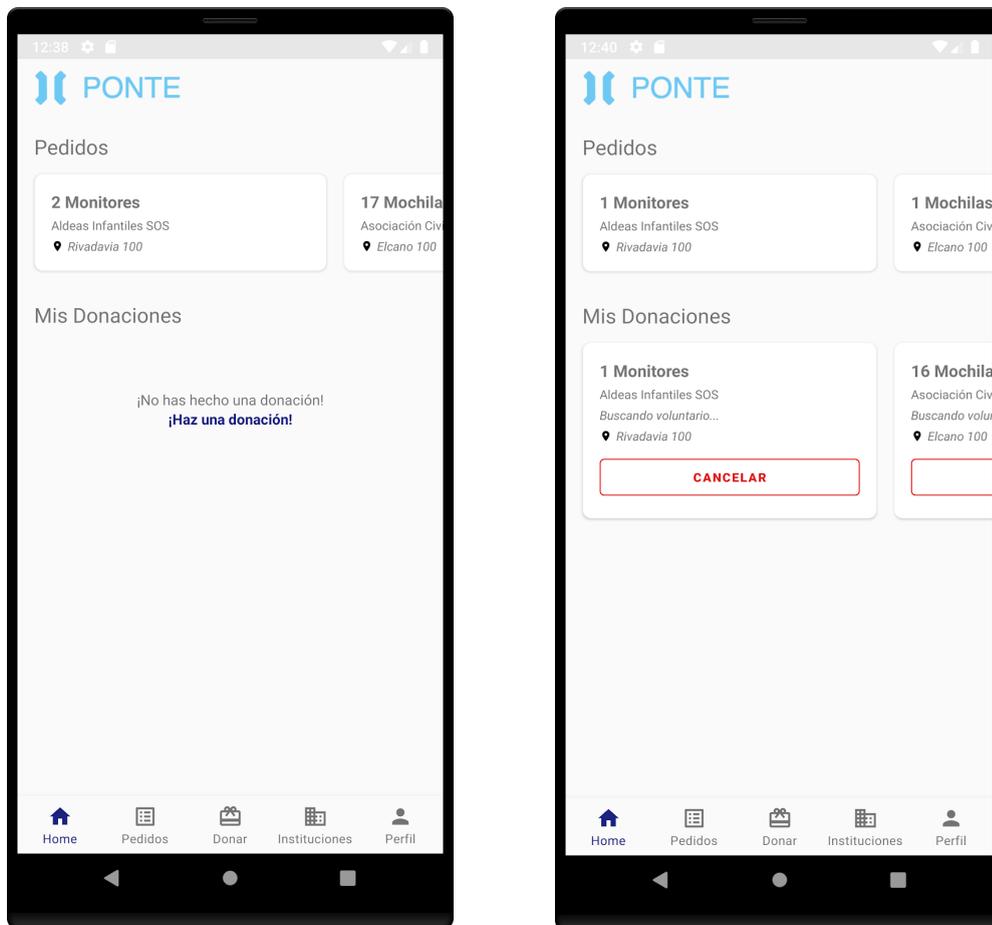


Figura 21: home de un donante, sin donaciones y con respectivamente

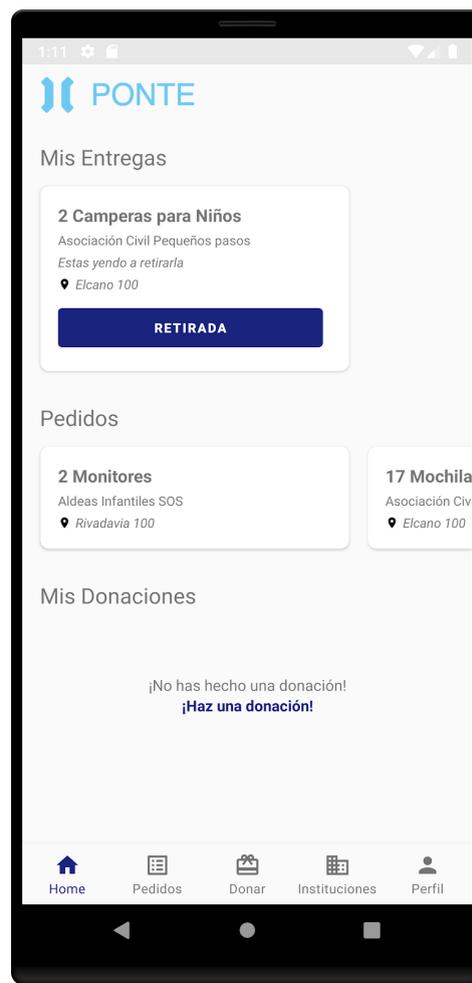
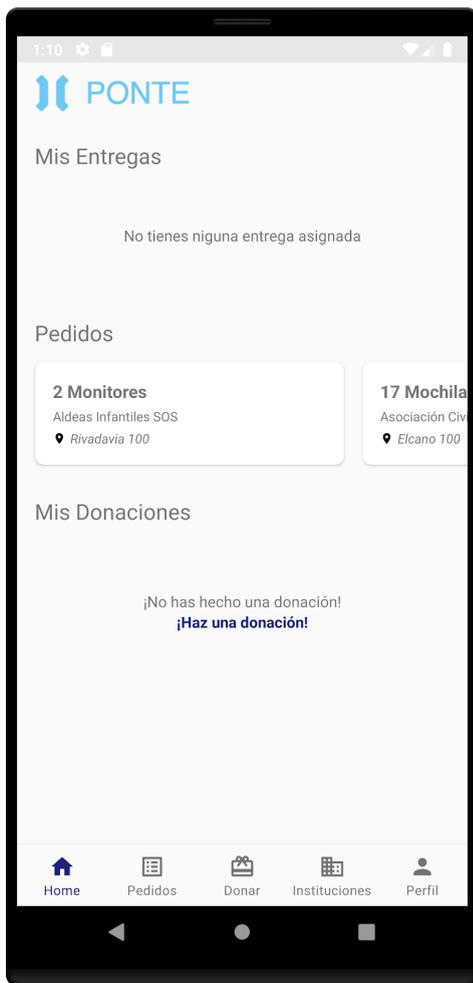


Figura 22: home de un voluntario sin y con entregas respectivamente

Pedidos e Instituciones

Las secciones de pedidos e instituciones son bastante sencillas, incluye únicamente un listado de los mismos (ver figura 23). En la sección de pedidos, uno puede seleccionar uno, para hacer una donación.

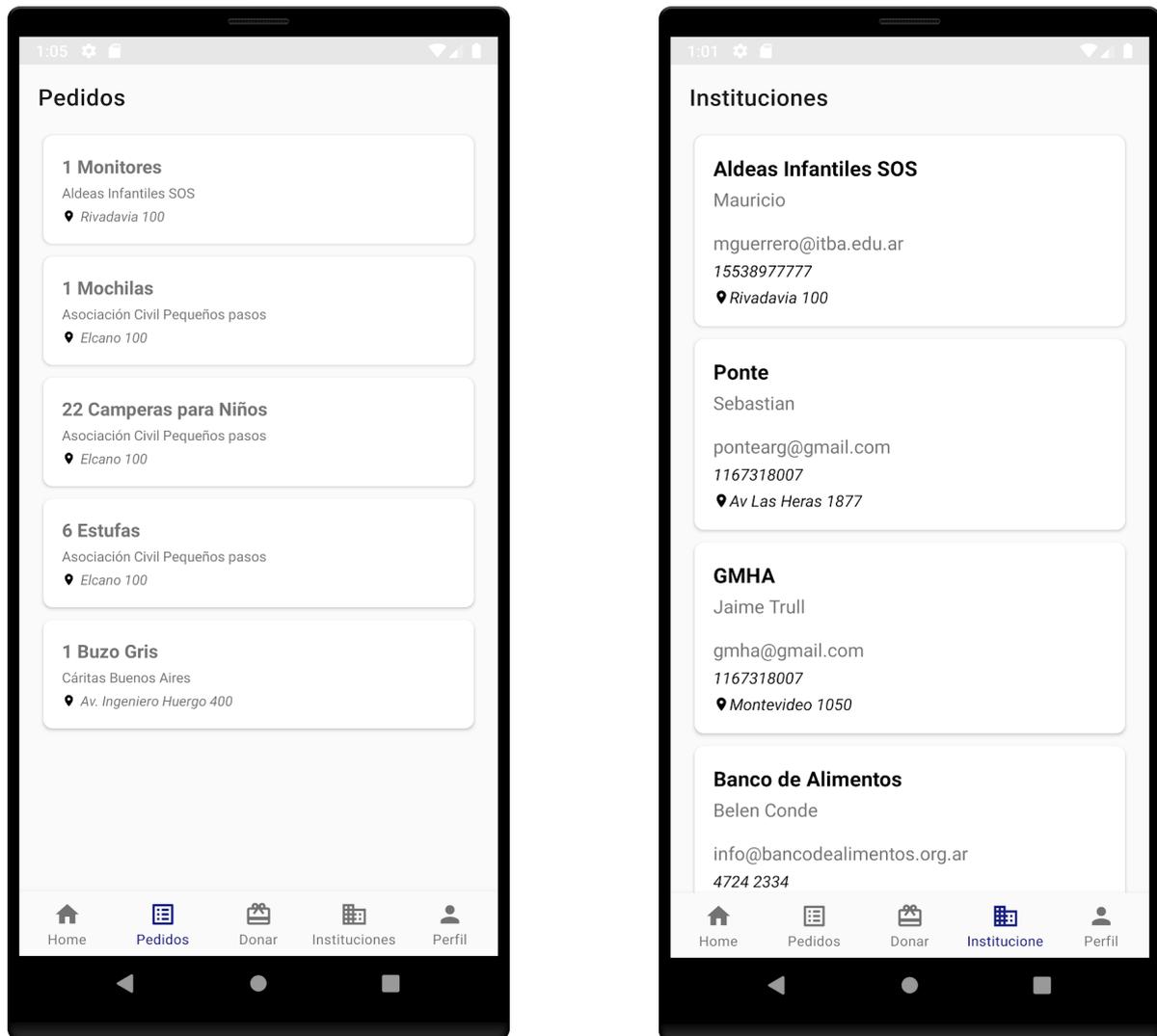


Figura 23: listados de pedidos e instituciones

Perfil

En el perfil varía dependiendo del estado del usuario. Si el usuario es un donante, se muestran sus donaciones y si es un voluntario, se le agregan sus entregas (Ver figura 24)

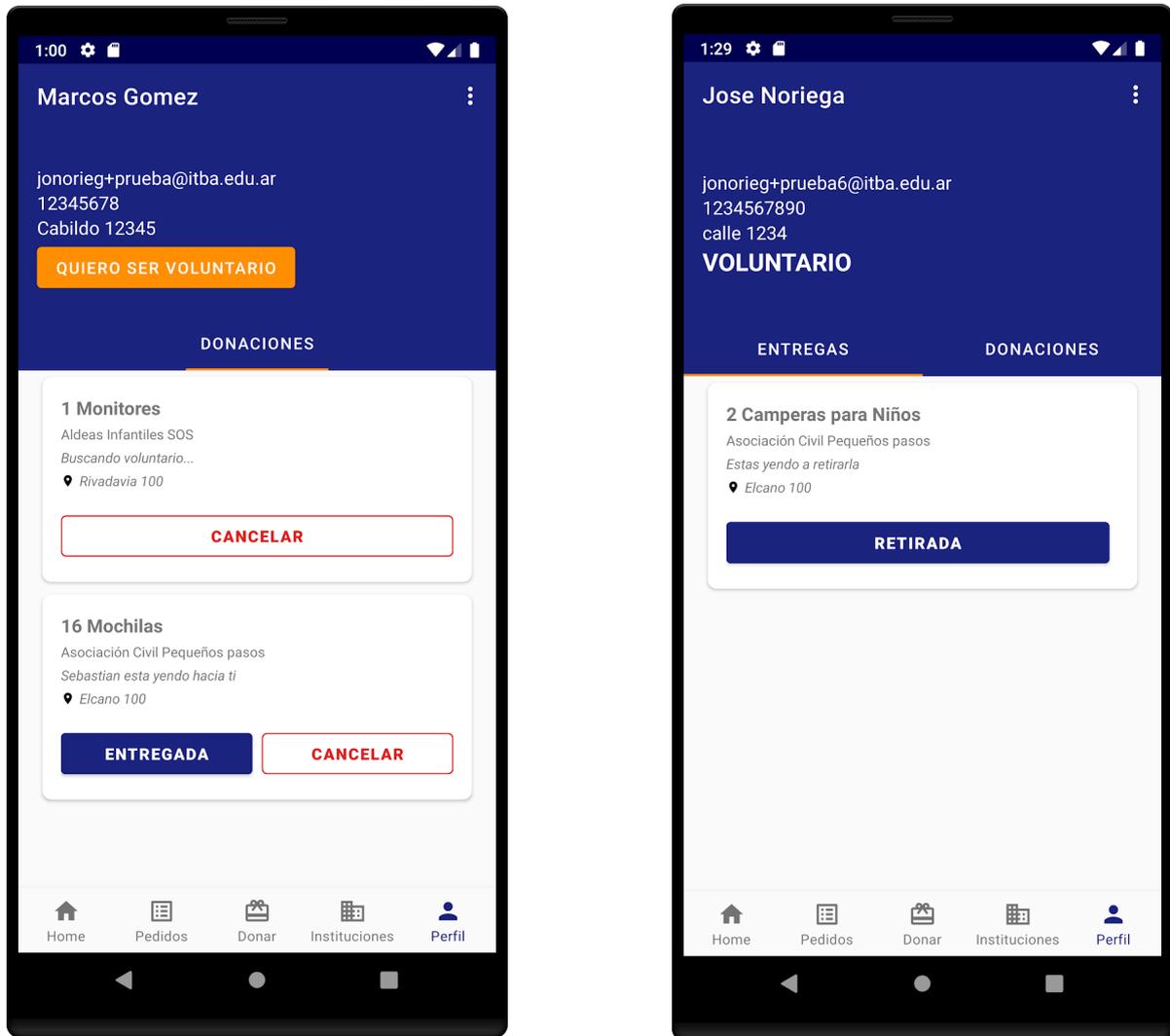


Figura 24: perfil de un donante y un voluntario respectivamente

Cuando el usuario desea ser voluntario, debe seleccionar “Quiero ser Voluntario” y luego completar un formulario. Una vez completado, deberá esperar a que un admin lo admita como en voluntario. (Ver figura 25 y 26)

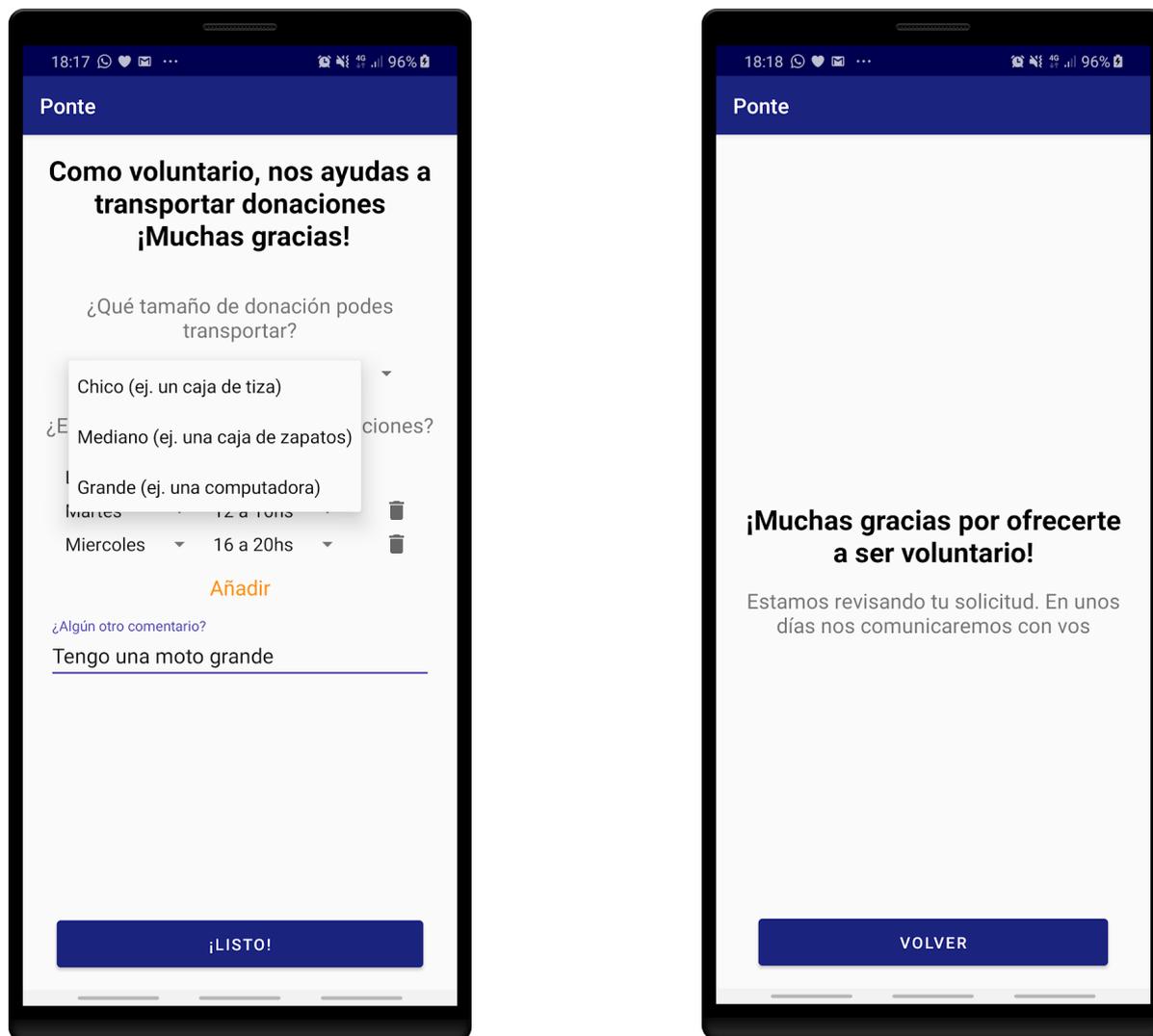


Figura 25: formulario para ser voluntario y confirmación del mismo

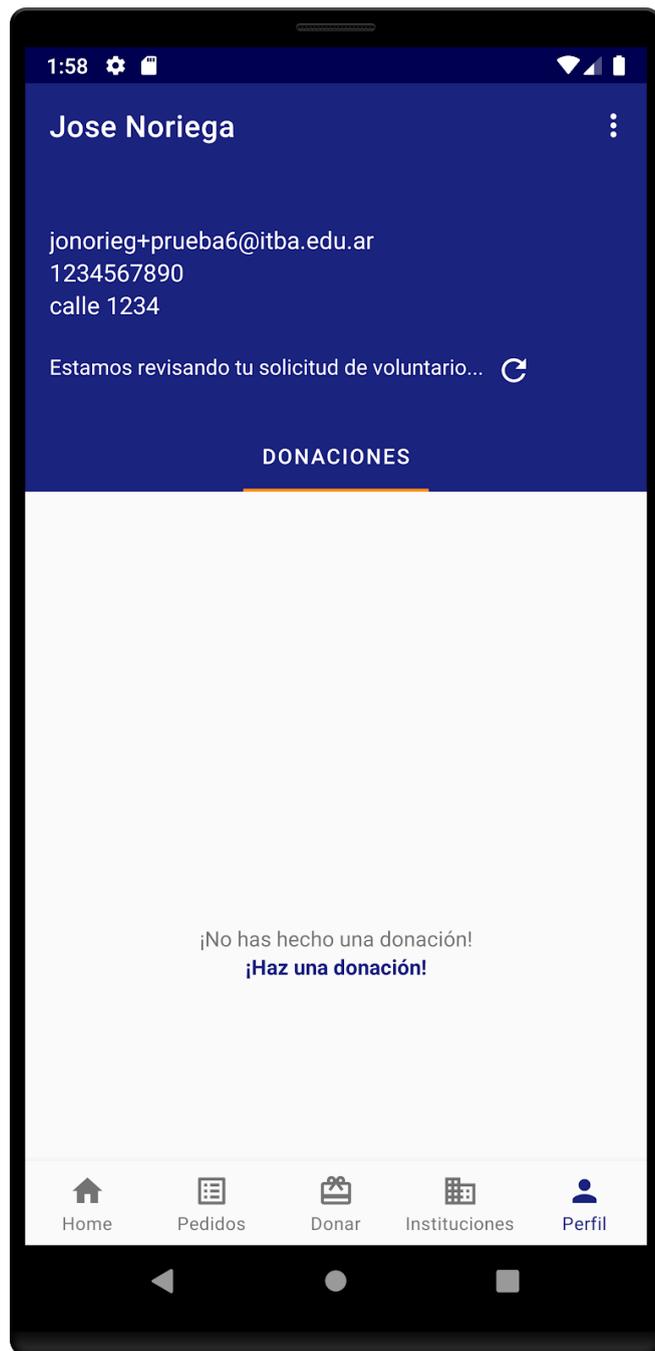


Figura 26: perfil de usuario que está esperando a convertirse en voluntario

Si selecciona el menú superior derecho se puede acceder al historial de entregas y donaciones. (Ver figura 27)

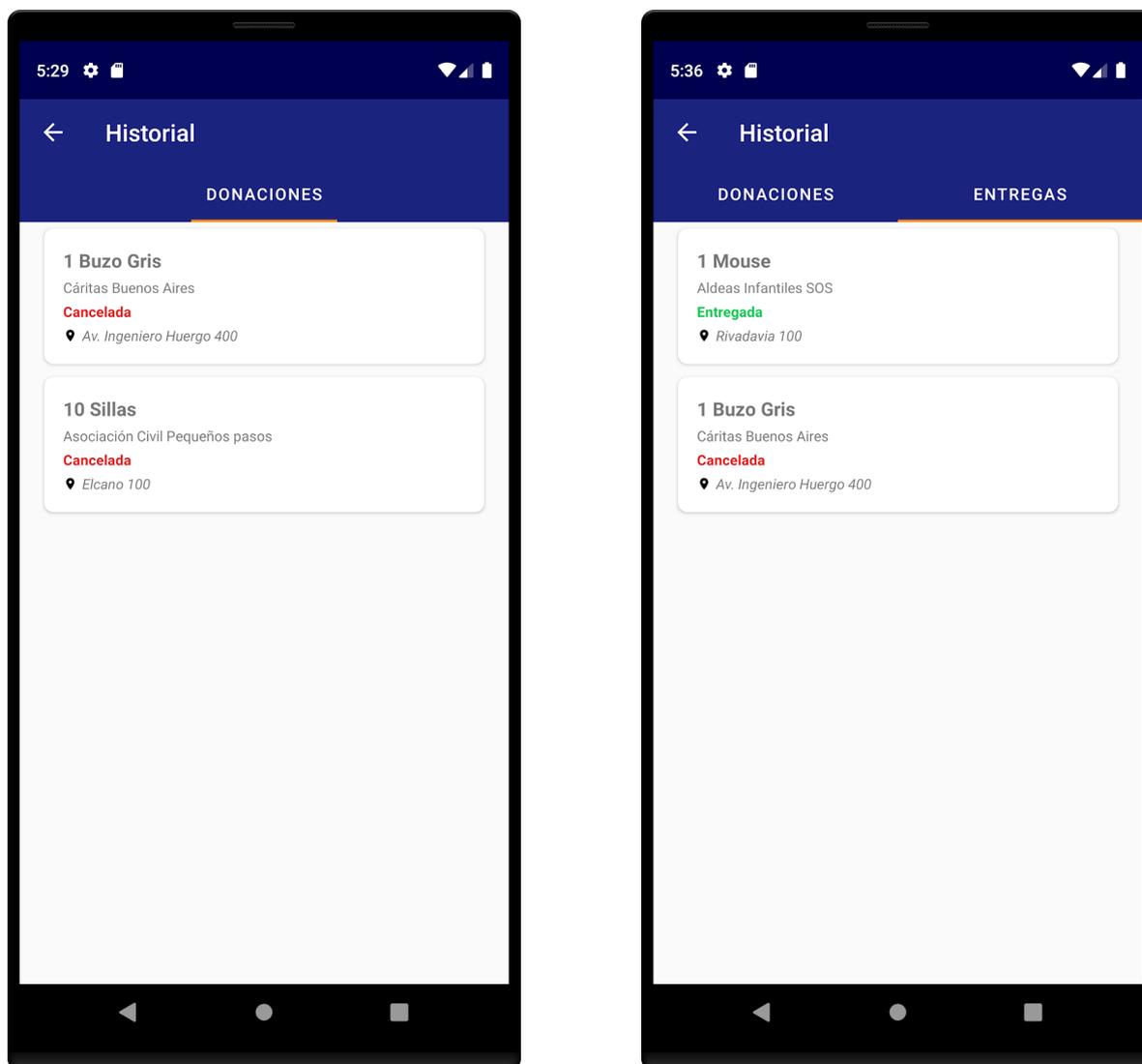


Figura 27: historial para una donante y un voluntario respectivamente

Donaciones

Como mencionamos anteriormente, un usuario puede hacer una donación nueva o puede donar a un pedido de una institución. Para realizar una nueva donación, se debe acceder al formulario desde el home. (Ver figura 28). También presenta errores: horarios repetidos, objeto de donación sin ingresar y cantidades inválidas (Ver figura 29)

The image displays two sequential screenshots of a mobile application's 'Donar' (Donate) form. Both screenshots show a top navigation bar with the time '1:04' and standard Android status icons. The bottom navigation bar includes icons for 'Home', 'Pedidos', 'Donar', 'Instituciones', and 'Perfil'. The 'Donar' icon is highlighted in blue in both screenshots.

The left screenshot shows the initial state of the form. It contains the following fields and elements:

- Header: 'Donar'
- Form container with a light gray border:
 - Text input: '¿Qué querés donar?'
 - Text input: '¿Cuánto querés donar?'
 - Text input: '¿En qué horario se puede retirar?' with a trash icon to its right.
 - Text input: 'Lunes 8 a 12hs' with an orange 'Añadir' button to its right.
 - Text input: '¿Algún otro comentario?'
- Bottom of form: A large orange button labeled 'DONAR'.

The right screenshot shows the form after some data has been entered:

- Text input: 'Mochila' (under '¿Qué querés donar?')
- Text input: '2' (under '¿Cuánto querés donar?')
- Text input: 'Lunes 8 a 12hs' with a trash icon to its right.
- Text input: 'Martes 12 a 16hs' with a trash icon to its right.
- Text input: 'Son azules' (under '¿Algún otro comentario?')
- Orange 'Añadir' buttons are present next to the schedule entries.

Figura 28: formulario de una nueva donación.

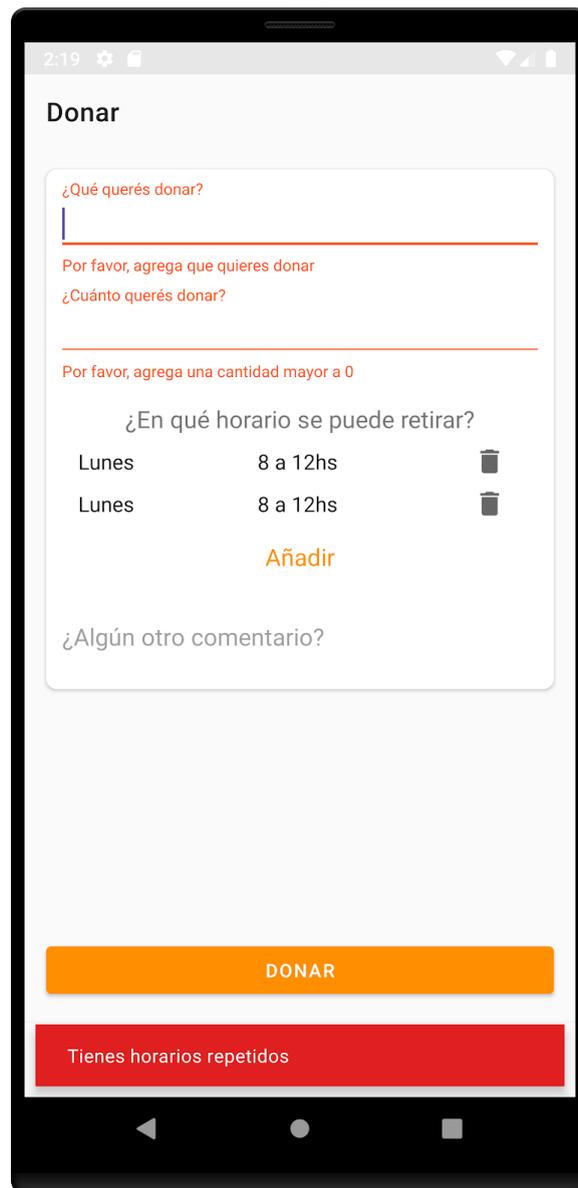


Figura 29: errores en el formulario de donación.

Si desea donar un pedido, solo debe seleccionar el mismo. Una vez que haya donado con éxito, se le notificará al usuario. A su vez se agrega un nuevo tipo de error: donar un monto mayor al pedido establecido. (Ver Figura 30)

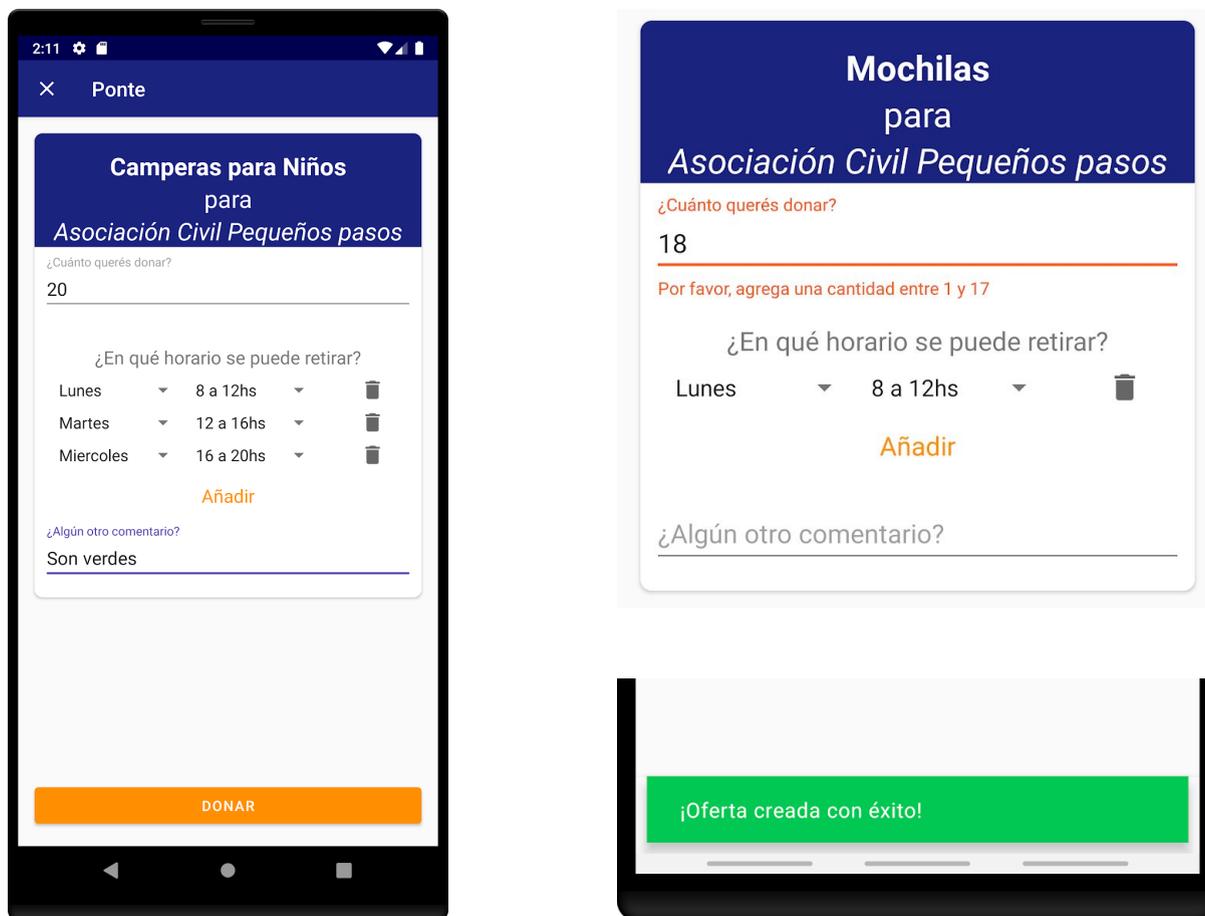


Figura 30: formulario de donación a un pedido hecho por una institución, con errores y confirmación

Estados de una donación

Estado: CREATED

Donante:

1 Buzo Gris
Buscando institución
Buscando voluntario...

CANCELAR

1 Buzo Gris
Cáritas Buenos Aires
Buscando voluntario...
Av. Ingeniero Huergo 400

CANCELAR

El estado CREATED es el estado inicial de una donación. Cuando el donante hace una oferta sin institución, verá el primer estado. En cambio si hace una donación a una institución, verá el segundo. Si selecciona CANCELAR, la donación pasará al estado CANCELLED.

Una vez que un voluntario haya sido asignado, la donación cambiará a VOLUNTEER-ASSIGNED

Estado: VOLUNTEER-ASSIGNED

Donante:

Voluntario:

1 Buzo Gris
Cáritas Buenos Aires
Jose es el voluntario asignado
Av. Ingeniero Huergo 400

CANCELAR

1 Buzo Gris
Cáritas Buenos Aires
Has sido asignado a esta donación
📍 Av. Ingeniero Huergo 400

IR A RETIRAR

Una vez que un voluntario fue asignado a la donación, el donante podrá ver quien es el mismo y a su vez podrá seguir teniendo la opción de cancelar. Desde el punto de vista del voluntario, le aparecerá la donación por primera vez, donde podrá marcar cuando efectivamente va a ir a retirarla. En ese momento la donación pasará a BEING-LOOKED-FOR

Estado: BEING-LOOKED-FOR

Donante:

1 Buzo Gris
Cáritas Buenos Aires
Jose esta yendo hacia ti
Av. Ingeniero Huergo 400

ENTREGADA **CANCELAR**

Voluntario:

1 Buzo Gris
Cáritas Buenos Aires
Estas yendo a retirarla
📍 Av. Ingeniero Huergo 400

RETIRADA

En este estado, se genera una bifurcación, dependiendo quien marque el estado primero. Esto se debe a que todavía no hay un backend que maneje internamente los estados.

Si el donante selecciona ENTREGADA antes que el voluntario (entregó la donación al mismo), el estado cambiará a DELIVERED-BY-DONOR.

Si el voluntario selecciona RETIRADA antes que el donante (retiro la donación del domicilio del donante), el estado cambiará a PICKED-BY-VOLUNTEER.

Estado: DELIVERED-BY-DONOR

Donante:

1 Buzo Gris

Cáritas Buenos Aires

Le entregaste la donación a Jose

Av. Ingeniero Huergo 400

Voluntario:

1 Buzo Gris

Cáritas Buenos Aires

El donante te entregó la donación

Av. Ingeniero Huergo 400

RETIRADA

Como se puede observar, desde el punto de vista del donante, se le ratifica que efectivamente entregó la donación al voluntario, anulando cualquier tipo de acción, debe esperar a que el voluntario presiona RETIRADA, y ahí pasará a BEING-TAKEN-TO.

Estado: PICKED-BY-VOLUNTEER

Donante:

1 Buzo Gris

Cáritas Buenos Aires

Jose marcó la donación como retirada

Av. Ingeniero Huergo 400

ENTREGADA

Voluntario:

1 Buzo Gris

Cáritas Buenos Aires

Has retirado la donación

Av. Ingeniero Huergo 400

Aquí es a la inversa del estado anterior, el voluntario se le informa que ha retirado la donación, y tiene que esperar a que el donante seleccione ENTREGADA para avanzar hacia BEING-TAKEN-TO.

Estado: BEING-TAKEN-TO

Donante:

1 Buzo Gris

Cáritas Buenos Aires

Jose esta en camino a entregarla

Av. Ingeniero Huergo 400

Voluntario:

1 Buzo Gris

Cáritas Buenos Aires

Estas en camino a entregarla

Av. Ingeniero Huergo 400

ENTREGADA

En este estado, el voluntario ya está llevando la donación al destino final. Al donante se le informa de eso, ya inhabilitado a realizar ninguna acción. Una vez que el voluntario entrega la donación, selecciona ENTREGADA, y la misma pasa a su estado final de DELIVERED.

Estado: DELIVERED o CANCELLED

Donante y Voluntario:

1 Mouse

Aldeas Infantiles SOS

Entregada

Rivadavia 100

1 Buzo Gris

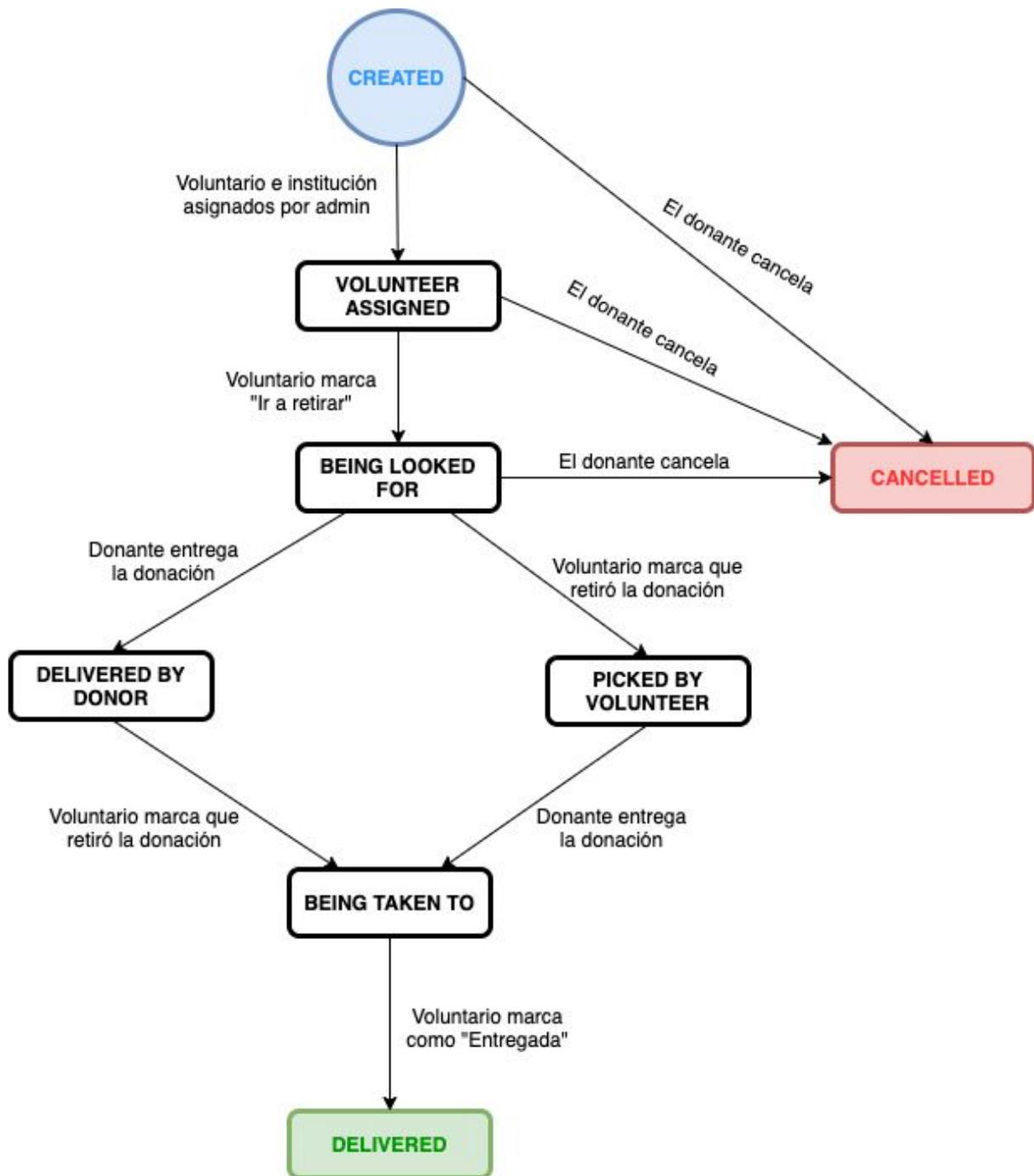
Cáritas Buenos Aires

Cancelada

Av. Ingeniero Huergo 400

En esta sección se ven los dos posibles estados finales de una donación, los cuales se pueden acceder yendo al menú de Historial, desde el perfil del usuario.

Gráfico completo del flujo de estados



Casos de Prueba

#Test	1		
Nombre	Crear una cuenta donante		
Objetivo	Un usuario cualquiera podrá ingresar a la aplicación y crear una cuenta para poder realizar donaciones		
Pre-condiciones	-		
Entradas	Datos de ingreso obligatorios: <ul style="list-style-type: none"> ● Nombre ● Apellido ● Email ● Contraseña ● Teléfono ● Dirección 		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra el formulario de inicio de sesión	Pasó
2	Se selecciona “¿No tienes cuenta? Únete a la causa”	Se muestra el formulario de registración	Pasó
3	Se cargan parcialmente los datos	Se muestra un error por cada dato faltante	Pasó
4	Se cargan datos utilizando un email ya utilizado	Se muestra un snackbar indicando el error	Pasó
5	Se cargan datos correctamente	Se muestra un mensaje de éxito y se le informa que debe confirmar su email	Pasó

#Test	2		
Nombre	Iniciar sesión en una cuenta existente		
Objetivo	Un usuario de Ponte puede ingresar a la aplicación con su cuenta creada previamente		
Pre-condiciones	Tener una cuenta de Ponte creada previamente		
Entradas	Datos de ingreso obligatorios: <ul style="list-style-type: none"> • Email • Contraseña 		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra el formulario de inicio de sesión	Pasó
2	Se cargan los datos utilizando un mail que no existe	Se muestra un error de usuario no registrado	Pasó
3	Se cargan un mail y una contraseña que no corresponde al mail	Se muestra un error diciendo que el email y/o la contraseña no coinciden	Pasó
4	Se cargan datos correctamente	Se ingresa a la sección de "Home" de la aplicación	Pasó

#Test	3		
Nombre	Ver los pedidos realizados por las instituciones		
Objetivo	Un usuario de Ponte puede ingresar a la aplicación a ver los pedidos		
Pre-condiciones	Haberse logueado en la app con una cuenta existente		
Entradas	-		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home	Pasó
2	Se selecciona "Pedidos" en la navegación inferior	Se muestra la pantalla con un listado de pedidos	Pasó

#Test	4		
Nombre	Ver las instituciones		
Objetivo	Un usuario de Ponte puede ingresar a la aplicación a ver las instituciones		
Pre-condiciones	Haberse logueado en la app con una cuenta existente		
Entradas	-		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home	Pasó
2	Se selecciona "Instituciones" en la navegación inferior	Se muestra la pantalla con un listado de las instituciones	Pasó

#Test	5		
Nombre	Donar un bien a un pedido		
Objetivo	Un usuario de Ponte puede ingresar a la aplicación a donar un bien que una institución esté pidiendo		
Pre-condiciones	Haberse logueado en la app con una cuenta existente		
Entradas	Datos de ingreso obligatorios: <ul style="list-style-type: none"> • Unidades a donar • Disponibilidad horaria Datos de ingreso opcional: <ul style="list-style-type: none"> • Comentarios adicionales 		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home	Pasó
2	Se selecciona un pedido desde el listado horizontal	Se muestra la pantalla con el formulario para donar con los datos de la donación pre cargados	Pasó
3	Se envía el formulario con una cantidad menor o igual a 0 o mayor a la cantidad solicitada	Se muestra un error indicando el rango de la cantidad que deben ingresar	Pasó
4	Se envía con una franja horaria repetida	Se muestra un error indicando que hay una franja horaria repetida	Pasó

5	Se envía con todos los datos correctos	Se muestra un mensaje de éxito, se lo redirige a la Home y se restan la cantidad donada al pedido	Pasó
---	--	---	------

#Test	6		
Nombre	Ofrecer un bien no pedido		
Objetivo	Un usuario de Ponte puede ingresar a la aplicación a donar un bien que nadie está pidiendo		
Pre-condiciones	Haberse logueado en la app con una cuenta existente		
Entradas	Datos de ingreso obligatorios: <ul style="list-style-type: none"> • Bien a donar • Unidades a donar • Disponibilidad horaria Datos de ingreso opcional: <ul style="list-style-type: none"> • Comentarios adicionales 		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home	Pasó
2	Se selecciona "Donar" en la navegación inferior	Se muestra la pantalla con el formulario para donar	Pasó
3	Se envía el formulario sin el bien a donar	Se muestra un error indicando que debe ingresar un bien	Pasó
4	Se envía el formulario con una cantidad menor o igual a 0 o mayor a la cantidad solicitada	Se muestra un error indicando el rango de la cantidad que deben ingresar	Pasó
5	Se envía con una franja horaria repetida	Se muestra un error indicando que hay una franja horaria repetida	Pasó
6	Se envía el formulario sin ningún dato cargado	Se muestra un error por cada campo obligatorio	Pasó
7	Se envía con todos los datos correctos	Se muestra un mensaje de éxito, se lo redirige a la Home y se restan la cantidad donada al pedido	Pasó

#Test	7		
Nombre	Ver las donación como donante y el estado de las mismas		
Objetivo	Un usuario de Ponte puede ingresar a la aplicación a ver el estado de sus donaciones		
Pre-condiciones	Haberse logueado en la app con una cuenta existente El usuario tiene que haber hecho donaciones		
Entradas	-		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home	Pasó
2	Se selecciona "Perfil" desde la navegación inferior	Se muestra el perfil con la información del usuario y las donaciones y sus respectivos estados	Pasó

#Test	8		
Nombre	Cancelar una donación		
Objetivo	Un usuario de Ponte puede ingresar a la aplicación y cancelar cualquier donación que haya realizado		
Pre-condiciones	Haberse logueado en la app con una cuenta existente El usuario tiene que haber hecho donaciones		
Entradas	-		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home con el listado de las donaciones activas	Pasó
2	Se selecciona "Cancelar" si el estado de la donación es el correspondiente	Se recarga la página y la donación no aparece en el listado. Pero aparece como cancelada en el historial de donaciones en la configuración del perfil	Pasó

#Test	9		
Nombre	Convertirse en Voluntario		
Objetivo	Un usuario de Ponte puede ser un voluntario además de un donante cuando lo desee		
Pre-condiciones	Haberse logueado en la app con una cuenta existente El usuario tiene que haber hecho donaciones		
Entradas	Datos de ingreso obligatorios: <ul style="list-style-type: none"> • Tamaño de vehículo • Franja horaria de disponibilidad Datos de ingreso opcional: <ul style="list-style-type: none"> • Comentarios adicionales Administrador revisa solicitud		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home	Pasó
2	Se selecciona "Perfil" en el menú de navegación inferior	Se muestra el perfil con la información del usuario y las donaciones y sus respectivos estados	Pasó
3	Se selecciona el "Quiero ser voluntario"	Se muestra el formulario de solicitud de voluntario	Pasó
4	Se completan los datos correctamente	Se muestra un agradecimiento	Pasó
5	Se selecciona "Volver"	Se vuelve al perfil y se indica "Estamos revisando tu solicitud de voluntario.."	Pasó
6	Una vez que el administrador aprueba la solicitud, se vuelve a entrar al perfil o se selecciona el botón de actualizar	Se muestra "VOLUNTARIO" en el perfil	Pasó

#Test	10		
Nombre	Ver listado de entregas		
Objetivo	Un voluntario desea ingresar a la aplicación y ver el listado de sus entregas pendientes		
Pre-condiciones	Haberse logueado en la app con una cuenta existente El usuario debe ser un voluntario		
Entradas	Administrador asigna voluntario a una donación		
#Paso	Descripción del paso	Resultado esperado	Pasó / Falló (Just.)
1	Se ingresa a la aplicación	Se muestra la Home con entregas vacías	Pasó
2	Administrador asigna al voluntario a una donación	Sin cambios	Pasó
3	El voluntario hace pull to refresh en la Home	Se muestra la entrega a la cual fue asignado.	Pasó

Mejoras Futuras

Debido a que esta es una primera aproximación a la versión Mobile de Ponte, naturalmente invita a pensar algunas mejoras, las cuales se volvieron evidentes durante el desarrollo.

Landing

Se tiene que agregar una landing o “onboarding” para cuando el usuario ingresa a la aplicación por primera vez, explicando el objetivo y cómo usar la aplicación de Ponte

Paginación

Todos los listados de instituciones, pedidos, entregas, etc. deberían paginarse, ya que si un listado se vuelve muy extenso, se tornaría muy ineficiente consultar el mismo además de que alentaría la aplicación. No es necesario traer todos los items del listado en un solo llamado, se pueden ir llamando a medida que se van necesitando.

Seguridad

La única seguridad que posee la aplicación por el momento es la contraseña del usuario. El sistema debería guardar sesiones en forma tokens, que tengan fecha de expiración. De esta manera, se podrían autenticar los endpoints, que por el momento son públicos. Esto se está siendo desarrollado en el proyecto final “Ponte 4” por alumnos del ITBA

Notifications

Es imperativo la implementación de notificaciones para el funcionamiento de la aplicación, ya que abarca varias funcionalidades. La principal es el anuncio de cambio de estado de una donación. Cada vez que cambie de estado, el usuario debería tener una notificación que le indique en que estado se encuentra la donación. Para ello, es necesario que el backend maneje los estados, y no el front end. Otro uso para la notificación, sería el de avisarle al usuario que su solicitud de voluntario ha sido aprobada y/o rechazada.

Geolocalización

La aplicación debería soportar geolocalización, tanto para el donante como para el voluntario. Desde el punto de vista del donante, se podría mostrar un mapa enseñándole por donde está el voluntario. Del lado del voluntario es más evidente, el mapa le irá indicando a donde se tiene que dirigir, siendo la casa del donante o el destino final. Debería poder utilizar el GPS

Donaciones

El backend no mantiene el estado de las donaciones, están son modificadas por los usuarios. Esto debería ser automático. De la misma forma, cuando se hace una donación, no se descuentan los ítems donados del pedido original, esto lo hace el front end tanto en web como en android. Debería manejarse desde el backend

Modo Offline

La aplicación podría soportar un modo offline, persistiendo la información en una base de datos local, para luego sincronizarla con el servidor

Conclusión

Uno de los desafíos del proyecto, fue seguir la línea de Ponte 1 tanto desde el punto de vista de arquitectura e integración, como de estética y usabilidad. Cumplir con los requisitos originales del proyecto, pero a su vez explotando las ventajas de la plataforma mobile, sin desviarse de la visión general de Ponte, pero agregando originalidad.

Si bien se cuenta con experiencia en Android, fue gracias al desarrollo de la aplicación, que se adquirió experiencia en la organización de un proyecto a mediano plazo. Dividir el desarrollo en relevamiento, diseño e implementación, permitió poner en práctica distintas herramientas aprendidas durante la carrera, para cada momento específico del desarrollo. Por eso, es oportuno detenerse y recordar el origen de dichas herramientas, que han permitido llevar a cabo todo el desarrollo del proyecto. Específicamente, nos referimos a las siguientes materias:

- Ingeniería de Software I
- Ingeniería de Software II
- Proyecto de Aplicaciones Web
- Human Computer Interface (HCI)
- Base de Datos I y II
- Gestión de Proyectos Informáticos

A su vez, se realizaron diversas reuniones con los alumnos que se encuentran desarrollando Ponte 3 y Ponte 4, y por supuesto uno de los creadores de Ponte 1, Sebastián Vera, para despejar dudas, compartir ideas, sugerir mejoras y plantear cambios.

Finalmente, nos interesa recuperar que fue a través del siguiente, que se toma conciencia del proceso por el cual atravesó el trabajo desde su comienzo hasta el final, que se entiende como un proyecto final de Ingeniería en Informática. Salió a la vista que PONTE es un proyecto ambicioso tanto tecnológica como socialmente. En los tiempos que se viven actualmente, es crucial que la tecnología juegue un rol central para conectar y ayudar a los

sectores postergados de la sociedad y PONTE tiene el potencial para ser el puente que conecta a los necesitados con gente que desea dar una mano. Esperemos que este proyecto sea uno de los primeros ladrillos en los cimientos de PONTE, y que sirva como puntapié e inspiración para futuros alumnos que se embarquen en el proyecto.